SMC with parameter learning

Hedibert Freitas Lopes

The University of Chicago Booth School of Business
5807 South Woodlawn Avenue, Chicago, IL 60637
http://faculty.chicagobooth.edu/hedibert.lopes

hlopes@ChicagoBooth.edu

Outline

Learning θ offline

Two-step strategy: On the first step, approximate $p(\theta|y^n)$ by

$$p^N(\theta|y^n) = \frac{p^N(y^n|\theta)p(\theta)}{p(y^n)} \propto p^N(y^n|\theta)p(\theta)$$

where $p^N(y^n|\theta)$ is a SMC approximation to $p(y^n|\theta)$. Then, on the 2nd step, sample θ via a MCMC scheme or a SIR scheme[1].

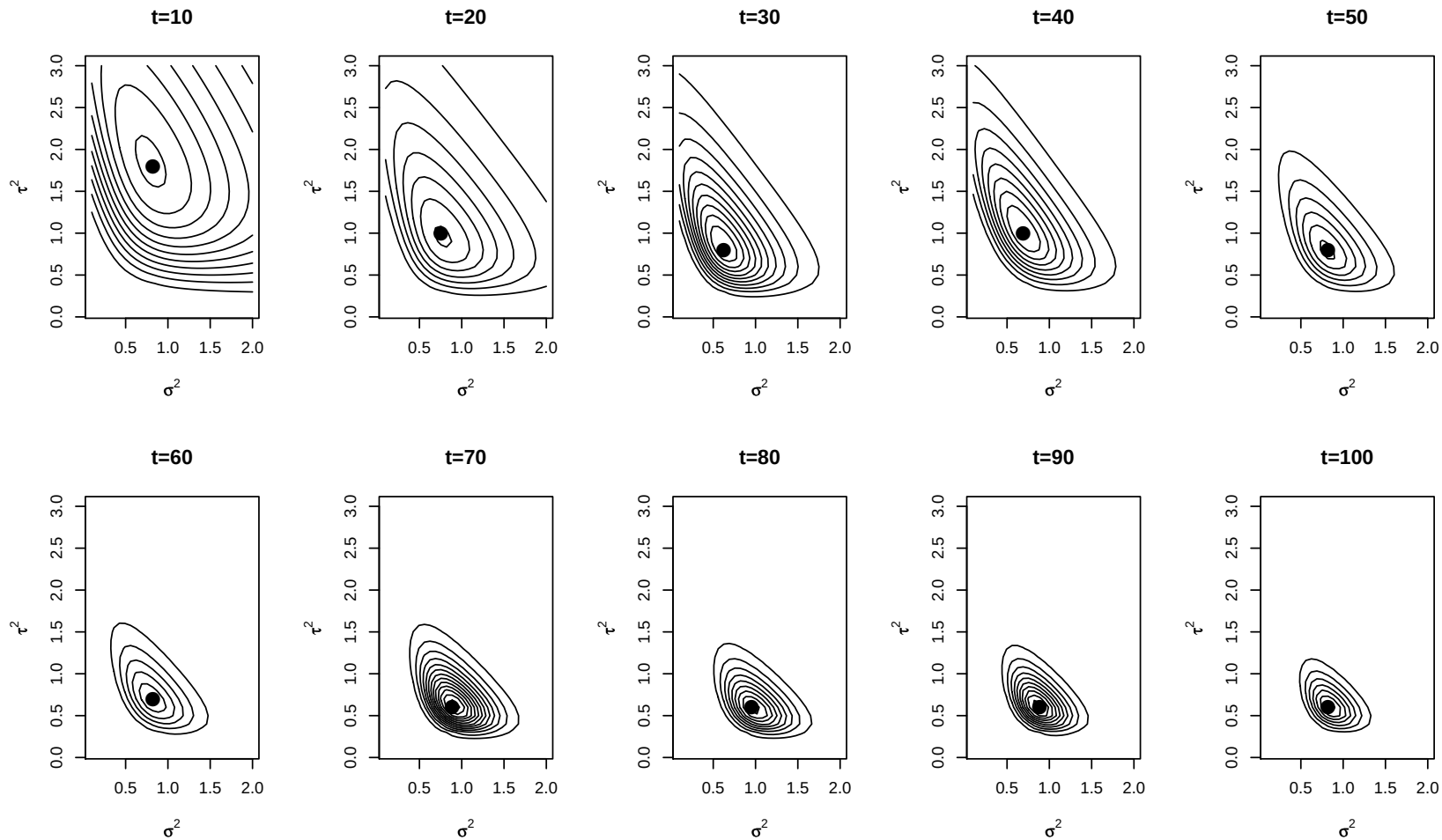
Problem 1: SMC looses its appealing sequential nature.

Problem 2: Overall sampling scheme is sensitive to $p^N(y|\theta)$.

[1] See Fernándes-Villaverde and Rubio-Ramírez (2007) "Estimating Macroeconomic Models: A Likelihood Approach", DeJong, Dharmarajan, Liesenfeld, Moura and Richard (2009) "Efficient Likelihood Evaluation of State-Space Representations" for applications of this two-step strategy to DSGE and related models.

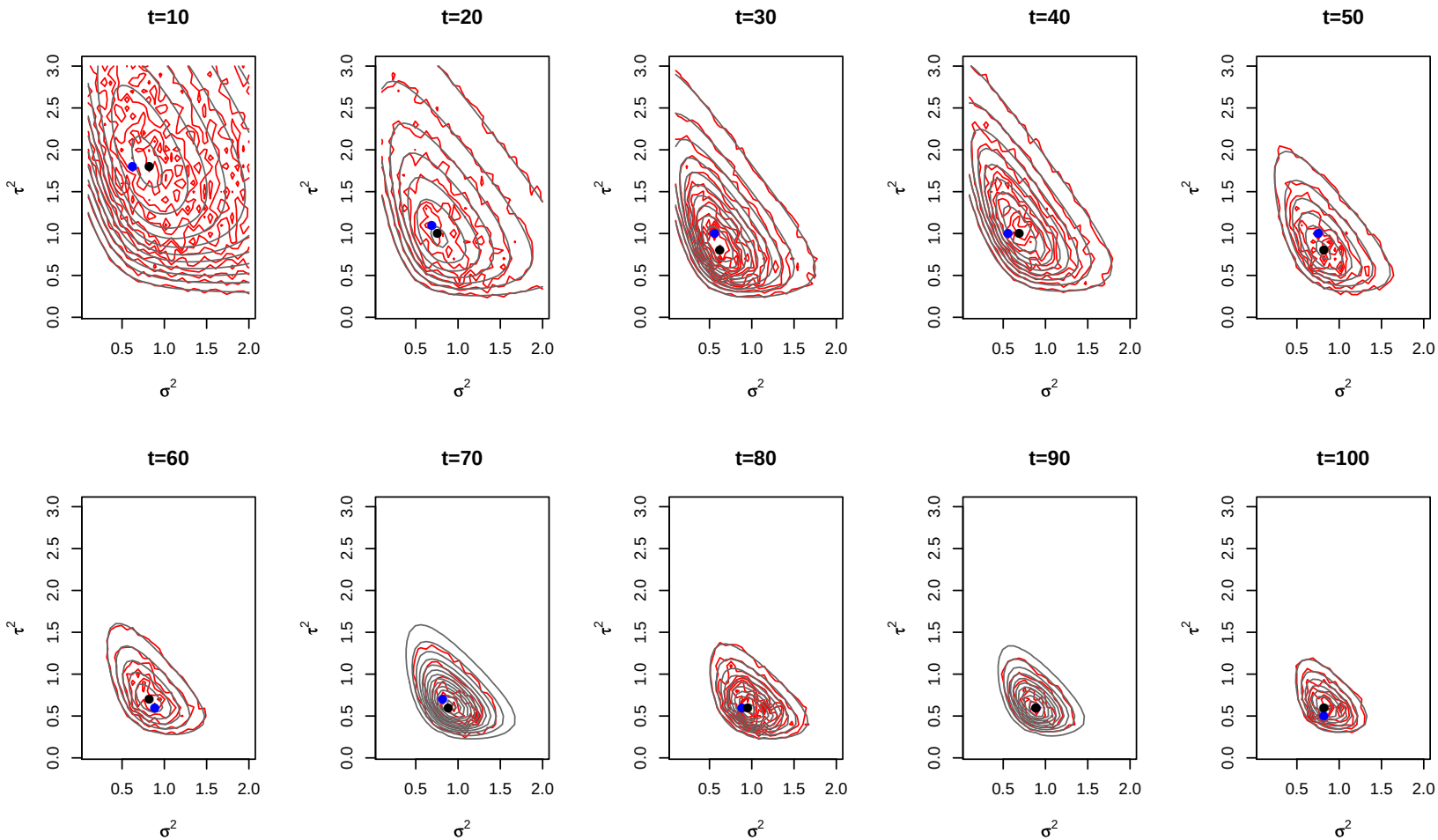
Example i: Exact integrated likelihood $p(y^n|\sigma^2, \tau^2)$

Let us revisit our 1st order DLM, where
$n = 100, x_0 = 0, \sigma^2 = 1, \tau^2 = 0.5$ and $x_0 \sim N(0.0, 100)$
30×30 grid: $\sigma^2 = (0.1, \ldots, 2)$ and $\tau^2 = (0.1, \ldots, 3)$



Example i: Approximated $p^N(y^n|\sigma^2, \tau^2)$

Based on $N = 1000$ particles



Learning θ sequentially

Sequentially learning x_t and θ.

$$\begin{aligned} \text{Posterior at t} &: p(x_t|\theta, y^t)p(\theta|y^t) \\ &\Downarrow \\ \text{Prior ate t+1} &: p(x_{t+1}|\theta, y^t)p(\theta|y^t) \\ &\Downarrow \\ \text{Posterior at t+1} &: p(x_{t+1}|\theta, y^{t+1})p(\theta|y^{t+1}) \end{aligned}$$

Advantages:

Sequential updates of $p(\theta|y^t)$, $p(x_t|y^t)$ and $p(\theta, x_t|y^t)$

Sequential h-steps ahead forecast $p(y_{t+h}|y^t)$

Sequential approximations for $p(y_t|y^{t-1})$

Sequential Bayes factors

$$B_{12t} = \frac{\prod_{j=1}^{t} p(y_j|y^{j-1}, M_1)}{\prod_{j=1}^{t} p(y_j|y^{j-1}, M_2)}$$

Liu and West filter

Liu and West (2001) approximates $p(\theta|y^t)$ by

$$p^N(\theta|y^t) = \sum_{i=1}^{N} \omega_t^{(i)} f_N(\theta|a\theta_t^{(i)} + (1-a)\bar{\theta}_t, (1-a^2)V_t)$$

where $\bar{\theta}_t$ and V_t approximate the mean and variance of θ, given y^t.

This leads to

$$p(\theta_{t+1}|x_t^{(i)}, \theta_t^{(i)}) = f_N(\theta_{t+1}|a\theta_t^{(i)} + (1-a)\bar{\theta}_t, (1-a^2)V_t)$$

and weights

$$\omega_{t+1}^{(i)} = \omega_t^{(i)} \frac{p(y_{t+1}|(x_{t+1}, \theta_{t+1})^{(i)})}{q_1((\tilde{x}_t, \tilde{\theta}_t)^{(i)})|y_{t+1})}.$$

Resampling step

$$q_1(x_t, \theta_t | y_{t+1}) = p(y_{t+1} | g(x_t), m(\theta_t))$$

where

$$\begin{aligned} g(x_t) &= E(x_{t+1} | x_t, m(\theta_t)) \\ m(\theta_t) &= a\theta_t + (1-a)\bar{\theta}_t \end{aligned}$$

The weights are then

$$\omega_{t+1}^{(i)} = \omega_t^{(i)} \frac{p(y_{t+1} | x_{t+1}^{(i)}, \theta_{t+1}^{(i)})}{p(y_{t+1} | g(\tilde{x}_t^{(i)}), m(\tilde{\theta}_t^{(i)}))}$$

Choice of a

Liu and West (2001) use a discount factor argument (see West and Harrison, 1997) to set the parameter a:

$$a = \frac{3\delta - 1}{2\delta}$$

For example,

- $\delta = 0.50$ leads to $a = 0.500$
- $\delta = 0.75$ leads to $a = 0.833$
- $\delta = 0.95$ leads to $a = 0.974$
- $\delta = 1.00$ leads to $a = 1.000$.

In the last case, i.e. $a = 1.0$, the particles of θ will degenerate over time to a single particle.

LW algorithm

For particles $\{(x_t, \theta_t, \omega_t)^{(j)}\}_{j=1}^N$ summarizing $p(x_t, \theta | y^t)$, estimates $\bar{\theta}_t = \sum_{i=1}^N \omega_t^{(i)} \theta_t^{(i)}$ and $V_t = \sum_{i=1}^N \omega_t^{(i)} (\theta_t^{(i)} - \bar{\theta}_t)(\theta_t^{(i)} - \bar{\theta}_t)'$, and given shrinkage parameter a, the algorithm runs as follows.

- For $i = 1, \ldots, N$, compute
  - $m(\theta_t^{(i)}) = a\theta_t^{(i)} + (1-a)\bar{\theta}_t$.
  - $g(x_t^{(i)}) = E(x_{t+1} | x_t^{(i)}, m(\theta_t^{(i)}))$.
  - $w_{t+1}^{(i)} = p(y_{t+1} | g(x_t^{(j)}), m(\theta_t^{(j)}))$.
- For $i = 1, \ldots, N$
  - Resample $(\tilde{x}_t, \tilde{\theta}_t)^{(i)}$ from $\{(x_t, \theta_t, w_{t+1})^{(j)}\}_{j=1}^N$.
  - Sample $\theta_{t+1}^{(i)} \sim N(m(\tilde{\theta}_t^{(i)}), h^2 V_t)$.
  - Sample $x_{t+1}^{(i)}$ from $p(x_{t+1} | \tilde{x}_t^{(i)}, \theta_{t+1}^{(i)})$.
  - Compute weight

$$\omega_{t+1}^{(i)} = \omega_t^{(i)} \frac{p(y_{t+1} | x_{t+1}^{(i)}, \theta_{t+1}^{(i)})}{p(y_{t+1} | g(\tilde{x}_t^{(i)}), m(\tilde{\theta}_t^{(i)}))}.$$

Example ii. State and parameter learning in the NDLM

Let us consider the following NDLM

$$\begin{aligned} y_t|x_t,\theta &\sim N(x_t,\sigma^2) \\ x_t|x_{t-1},\theta &\sim N(\alpha+\beta x_{t-1},\tau^2) \end{aligned}$$

with $x_0 \sim N(m_0, C_0)$ and $\theta = (\alpha, \beta, \sigma^2, \tau^2)$.

The optimal resampling distribution is

$$(y_t|x_{t-1},\theta) \sim N(\alpha+\beta x_{t-1},\sigma^2+\tau^2).$$

The optimal sampling distributions is

$$(x_t|x_{t-1},y^t,\theta) \sim N((1-A)(\alpha+\beta x_{t-1})+Ay_t, A\sigma^2),$$

where $A = \tau^2/(\sigma^2+\tau^2)$.

Example ii. Learning θ

Assume that the prior of $\theta = (\alpha, \beta, \tau^2, \sigma)$ is

$$p(\theta|s_0) = p_{IG}(\sigma^2; n_0/2, n_0\sigma_0^2/2)p_{NIG}(\gamma, \tau^2; g_0, G_0, \nu_0/2, \nu_0\tau_0^2/2),$$

where $\gamma = (\alpha, \beta)$ and known $s_0 = (n_0, \sigma_0^2, g_0, G_0, \nu_0, \tau_0^2)$.

It follows that

$$p(\theta|s_t) = p_{IG}(\sigma^2; n_t/2, n_t\sigma_t^2/2)p_{NIG}(\gamma, \tau^2; g_t, G_t, \nu_t/2, \nu_t\tau_t^2/2),$$

where $n_t = n_{t-1} + 1$, $\nu_t = \nu_{t-1} + 1$, $z_t = (1, x_{t-1})'$,

$$\begin{aligned} n_t\sigma_t^2 &= n_{t-1}\sigma_{t-1}^2 + (y_t - x_t)^2 \\ G_t^{-1} &= G_{t-1}^{-1} + z_t z_t' \\ G_t^{-1}g_t &= G_{t-1}^{-1}g_{t-1} + z_t x_t \\ \nu_t\tau_t^2 &= \nu_{t-1}\tau_{t-1}^2 + x_t^2 - g_t' G_t^{-1} g_t \end{aligned}$$

and

$$s_t = (n_t, \sigma_t^2, g_t, G_t, \nu_t, \tau_t^2) = \mathcal{S}(s_{t-1}, x_{t-1}, x_t, y_t).$$

Particle learning

For particle set $\{(x_{t-1}, s_{t-1}, \theta)^{(i)}\}_{i=1}^{N}$, the algorithm is:

1. Resample $(\tilde{x}_{t-1}, \tilde{s}_{t-1}, \tilde{\theta})^{(i)}$ from the above set with weights

$$\omega_t^{(i)} \propto p(y_t | x_{t-1}^{(i)}, \theta^{(i)});$$

2. Sample $x_t^{(i)} \sim p(x_t | \tilde{x}_{t-1}^{(i)}, y^t, \tilde{\theta}^{(i)})$;
3. Update $s_t^{(i)} = \mathcal{S}(\tilde{s}_{t-1}^{(i)}, \tilde{x}_{t-1}^{(i)}, x_t^{(i)}, y_t)$;
4. Sample $\theta^{(i)} \sim p(\theta | s_t^{(i)})$.

Storvik's filter

For particle set $\{(x_{t-1}, s_{t-1}, \theta)^{(i)}\}_{i=1}^{N}$, the algorithm is:

1. Sample $x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, y^t, \theta^{(i)})$;
2. Resample $(\tilde{x}_{t-1}, \tilde{x}_t, \tilde{s}_{t-1}, \tilde{\theta})^{(i)}$ from the above set with weights

$$\omega_t^{(i)} \propto p(y_t | x_{t-1}^{(i)}, \theta^{(i)});$$

3. Update $s_t^{(i)} = \mathcal{S}(\tilde{s}_{t-1}^{(i)}, \tilde{x}_{t-1}^{(i)}, \tilde{x}_t^{(i)}, y_t)$;
4. Sample $\theta^{(i)} \sim p(\theta | s_t^{(i)})$;
5. Set $x_t^{(i)} = \tilde{x}_t^{(i)}$.

See Storvik (2002) and Fearnhead (2002).

Integrating x_{t-1} out

Let $(x_{t-1}|y^{t-1}, \theta) \equiv (x_{t-1}|r_{t-1}, \theta) \sim N(m_{t-1}, C_{t-1})$, where

$$r_{t-1} = (m_{t-1}, C_{t-1}).$$

Goal: $r_t = \mathcal{R}(r_{t-1}, \theta)$.

The optimal resampling distribution is

$$(y_t|y^{t-1}, \theta) \equiv (y_t|r_{t-1}, \theta) \sim N(a_t, Q_t)$$

where $a_t = \alpha + \beta m_{t-1}$ and $Q_t = \beta^2 C_{t-1} + \tau^2 + \sigma^2$.

It is easy to see that

$$(x_t|y^t, \theta) \equiv (x_t|y_t, r_{t-1}, \theta) \sim N(m_t, C_t)$$

where $m_t = (1 - A_t)a_t + A_t y_t$ and $C_t = A_t \sigma^2$, for $A_t = R_t/Q_t$.

However, in order to update s_t (and sample θ) we need to sample

$$(x_{t-1}, x_t) \sim p(x_{t-1}, x_t | y_t, r_{t-1}, \theta)$$

It can be shown that

$$(x_t | x_{t-1}, y_t, r_{t-1}, \theta) \sim N((1-A)(\alpha + \beta x_{t-1}) + A y_t, A\sigma^2)$$

and

$$(x_{t-1} | y_t, r_{t-1}, \theta) \sim N(v_x, V_x)$$

where

$$\begin{aligned} A &= \tau^2/(\sigma^2 + \tau^2) \\ V_x^{-1} &= C_{t-1}^{-1} + A\tau^{-2}\beta^2 \\ V_x^{-1} v_x &= C_{t-1} m_{t-1} + A\tau^{-2}\beta(y_t - \alpha) \end{aligned}$$

PL with state sufficient statistics

For particle set $\{(r_{t-1}, s_{t-1}, \theta)^{(i)}\}_{i=1}^{N}$, the algorithm is:

1. Resample $(\tilde{r}_{t-1}, \tilde{s}_{t-1}, \tilde{\theta})^{(i)}$ from the above set with weights
$$\omega_t^{(i)} \propto p(y_t | r_{t-1}^{(i)}, \theta^{(i)});$$
2. Sample $x_{t-1}^{(i)} \sim p(x_{t-1} | y_t, \tilde{r}_{t-1}^{(i)}, \tilde{\theta}^{(i)})$;
3. Sample $x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, y_t, \tilde{r}_{t-1}^{(i)}, \tilde{\theta}^{(i)})$;
4. Update $s_t^{(i)} = \mathcal{S}(\tilde{s}_{t-1}^{(i)}, x_{t-1}^{(i)}, x_t^{(i)}, y_t)$;
5. Sample $\theta^{(i)} \sim p(\theta | s_t^{(i)})$;
6. Update $r_t^{(i)} = \mathcal{R}(\tilde{r}_{t-1}^{(i)}, \tilde{\theta}^{(i)})$.

Example ii. Bootstrap filter with learning θ

For particle set $\{(x_{t-1}, s_{t-1}, \theta)^{(i)}\}_{i=1}^{N}$, the algorithm is:

1. Sample $\tilde{x}_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, \theta^{(i)})$;
2. Sample $k^i \sim \{1, \ldots, M\}$ with $\omega_t^{(i)} \propto p(y_t | \tilde{x}_t^{(i)})$;
3. Set $x_t^{(i)} = \tilde{x}_t^{(k^i)}$;
4. Update $s_t^{(i)} = \mathcal{S}(s_{t-1}^{(k^i)}, x_{t-1}^{(k^i)}, x_t^{(i)}, y_t)$;
5. Sample $\theta^{(i)} \sim p(\theta | s_t^{(i)})$.

Example ii. Auxiliary particle filter with learning θ

For particle set $\{(x_{t-1}, s_{t-1}, \theta)^{(i)}\}_{i=1}^{N}$, the algorithm is:

1. Resample $(\tilde{x}_{t-1}, \tilde{s}_{t-1}, \tilde{\theta})^{(i)}$ from the above set with weights

$$\omega_t^{(i)} \propto p(y_t | g(x_{t-1}^{(i)}), \theta^{(i)});$$

2. Sample $\tilde{x}_t^{(i)} \sim p(x_t | \tilde{x}_{t-1}^{(i)}, \tilde{\theta}^{(i)})$;
3. Sample $k^i \sim \{1, \ldots, M\}$ with

$$\pi_t^{(i)} \propto p(y_t | \tilde{x}_t^{(i)}, \tilde{\theta}^{(i)}) / \omega_t^{(k^i)};$$

4. Set $x_t^{(i)} = \tilde{x}_t^{(k^i)}$;
5. Update $s_t^{(i)} = \mathcal{S}(\tilde{s}_{t-1}^{(k^i)}, \tilde{x}_{t-1}^{(k^i)}, x_t^{(i)}, y_t)$;
6. Sample $\theta^{(i)} \sim p(\theta | s_t^{(i)})$.

Example ii. Comparison between LWF, SF and PL

$T = 200$ obs. simulated from $\theta = (0.0, 0.9, 0.5, 1.0)$ and $x_0 = 0$.

The prior hyperparameters are $m_0 = 0$, $C_0 = 10$, $g_0 = (0.0, 0.9)'$, $G_0 = I_2$, $n_0 = \nu_0 = 10$, $\tau_0^2 = 0.5$ and $\sigma_0^2 = 1.0$.

Each $N = 1000$ particle filter is replicated $R = 100$ times.

A very long PL ($N = 100000$) is run to serve as a benchmark for comparison.

Let $q(\gamma, \alpha, t)$ be the 100αth percentile of $p(\gamma|y^t)$, where γ is an element of θ. We define the root mean squared error as the square root of

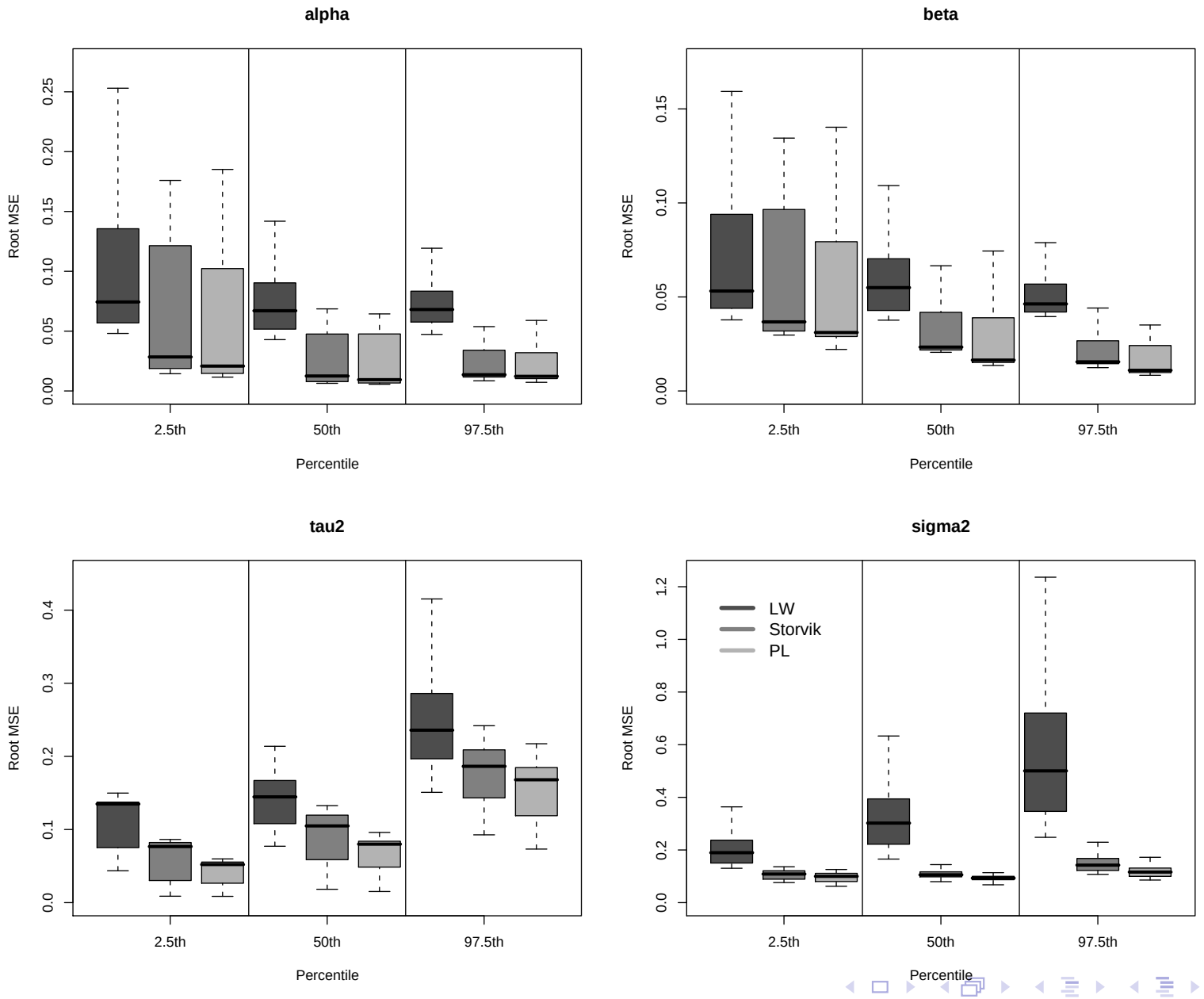
$$MSE(\gamma, \alpha, f, t) = \sum_{t,r} [q(\gamma, \alpha, t) - q_{fr}(\gamma, \alpha, t)]^2 / R$$

for filter f in $\{$LW,STORVIK,PL$\}$ and replication $r = 1, \ldots, R$.

All filters are fully adapted.

- LW differs from PL only through the estimation of θ.
- Storvik: sample-resample
- PL: resample-sample

PL and SF are significantly better than the LWF.
PL is moderately better than SF.



Example iii. Sample-resample or PL?

Three time series of length $T = 1000$ were simulated from

$$\begin{aligned} y_t|x_t, \sigma^2 &\sim N(x_t, \sigma^2) \\ x_t|x_{t-1}, \tau^2 &\sim N(x_{t-1}, \tau^2) \end{aligned}$$

with $x_0 = 0$ and (σ^2, τ^2) in $\{(0.1, 0.01), (0.01, 0.01), (0.01, 0.1)\}$. Throughout σ^2 is kept fixed.

The independent prior distributions for x_0 and τ^2 are $x_0 \sim N(m_0, V_0)$ and $\tau^2 \sim IG(a, b)$, for $a = 10$, $b = (a+1)\tau_0^2$, $m_0 = 0$ and $V_0 = 1$, where τ_0^2 is the true value of τ^2 for a given study.

We also include BBF in the comparison, for completion.

In all filters τ^2 is sampled offline from $p(\tau^2|S_t)$ where S_t is the vector of conditional sufficient statistics.

Example iii. Mean absolute error

The three filters are rerun $R = 100$ times, all with the same seed within run, for each one of the three simulated data sets. Five different number of particles N were considered: 250, 500, 1000, 2000 and 5000.

Mean absolute errors (MAE) taken over the 100 replications are constructed by comparing percentiles of the true sequential distributions $p(x_t|y^t)$ and $p(\tau^2|y^t)$ to percentiles of the estimated sequential distributions $p_N(x_t|y^t)$ and $p_N(\tau^2|y^t)$.

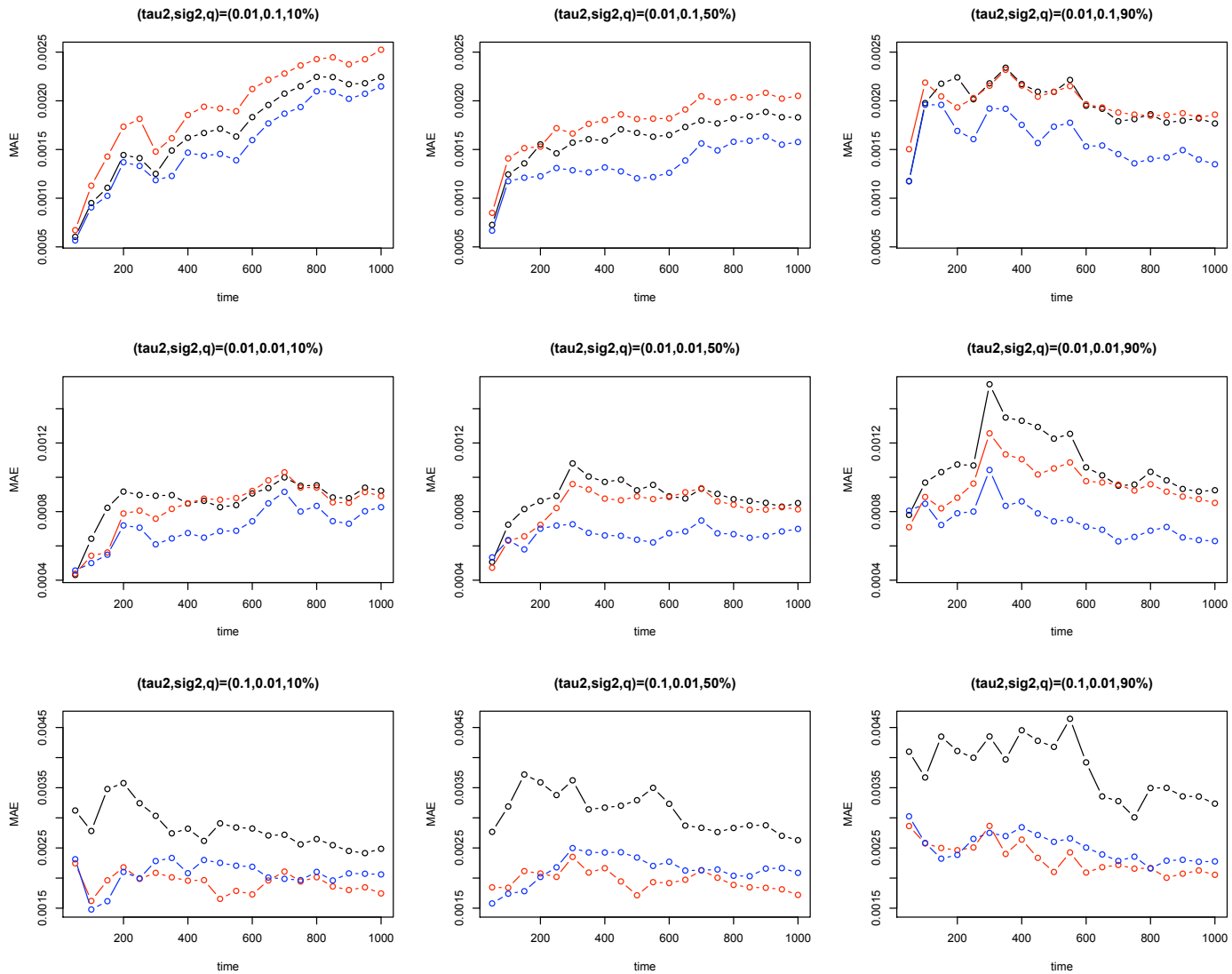
For $\alpha = 0.1, 0.5, 0.9$, true and estimated values of $q^x_{t,\alpha}$ and $q^{\tau^2}_{t,\alpha}$ were computed, for $Pr(x_t < q^x_{t,\alpha}|y^t) = Pr(\tau^2 < q^{\tau^2}_{t,\alpha}|y^t) = \alpha$.

For a in $\{x, \tau^2\}$ and α in $\{0.01, 0.50, 0.99\}$,

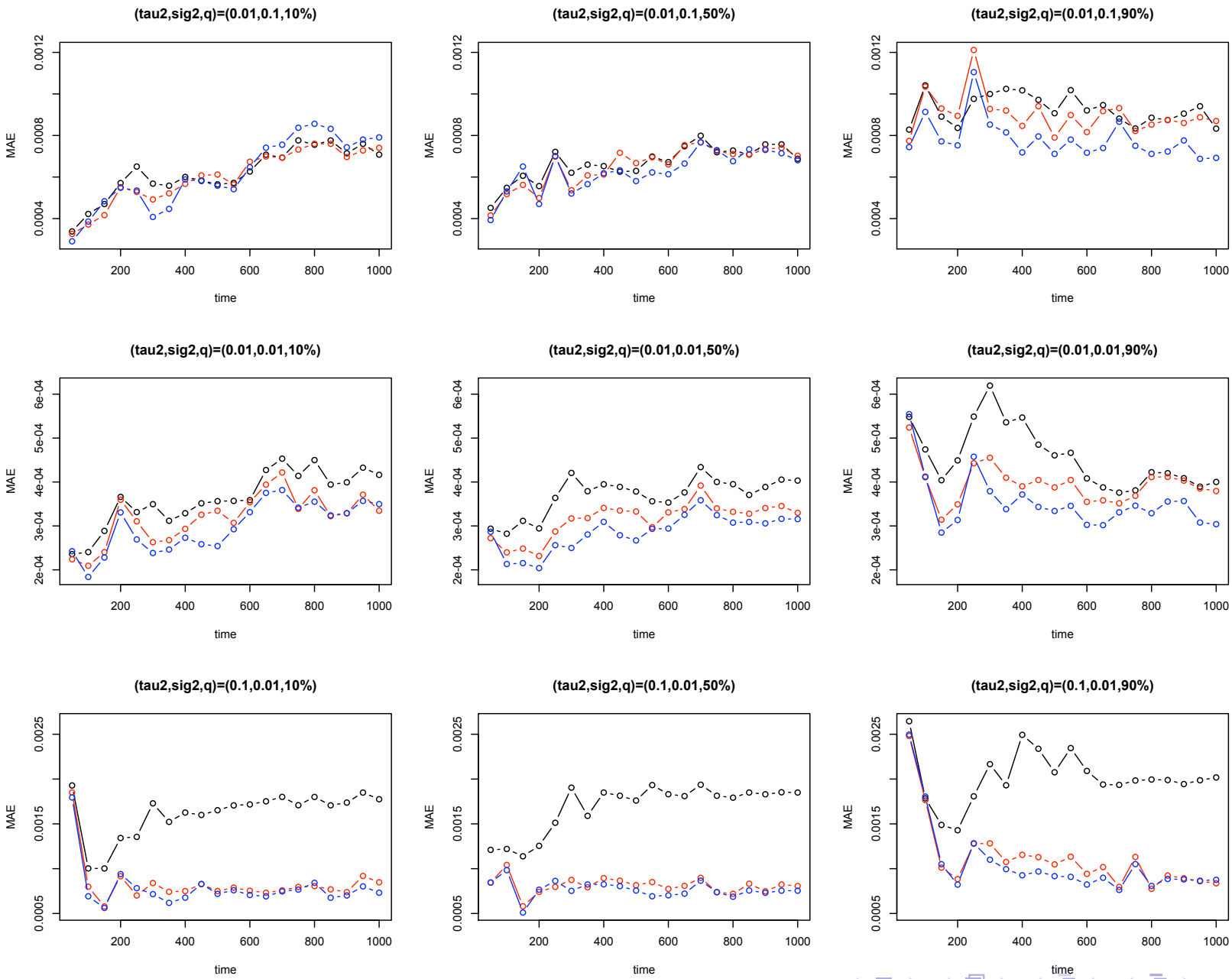
$$MAE^a_{t,\alpha} = \frac{1}{R}\sum_{r=1}^{R} |q^a_{t,\alpha} - \hat{q}^a_{t,\alpha,r}|$$

Example iii. $M = 500$ and learning τ^2.

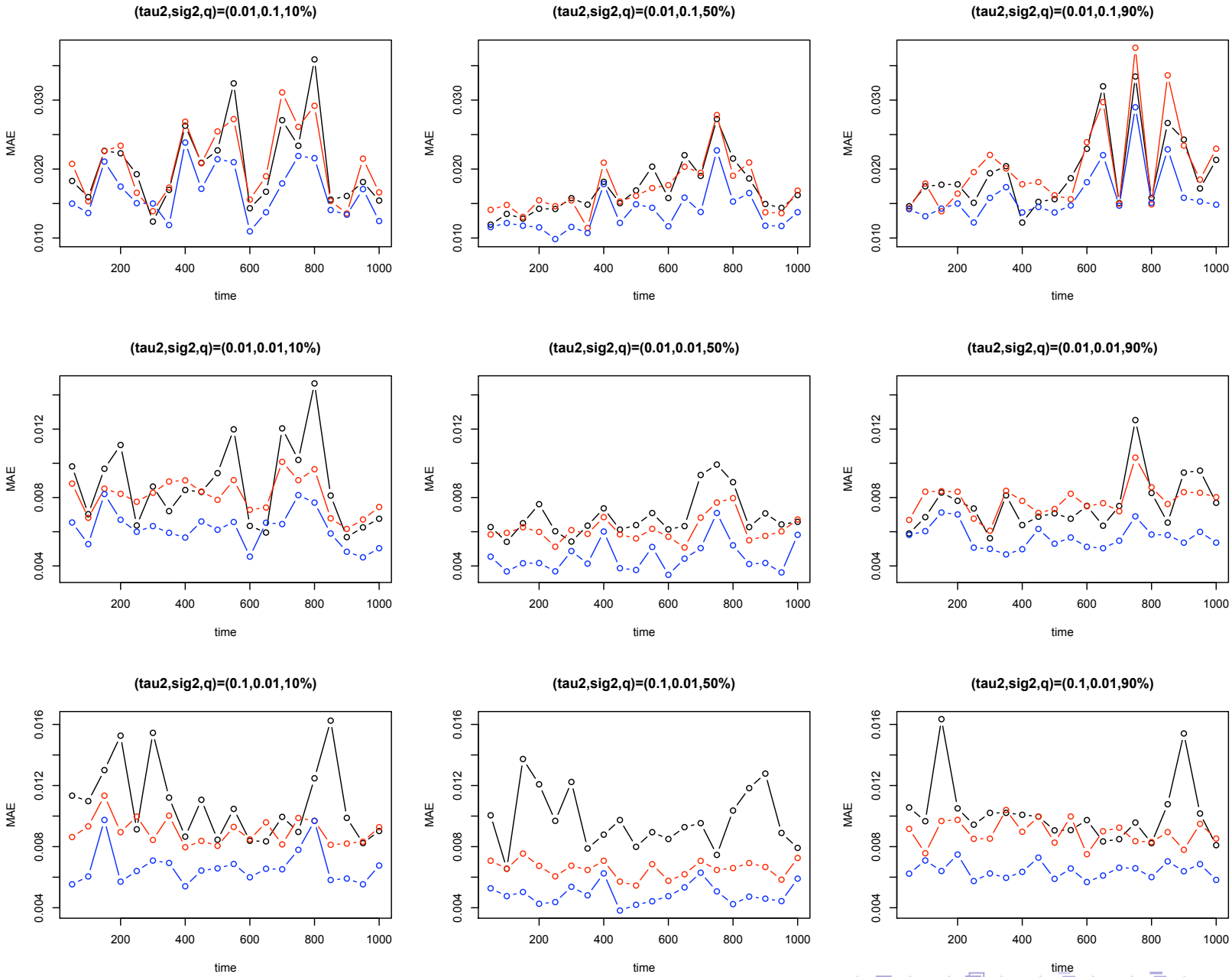
BBF,sample-resample,PL.



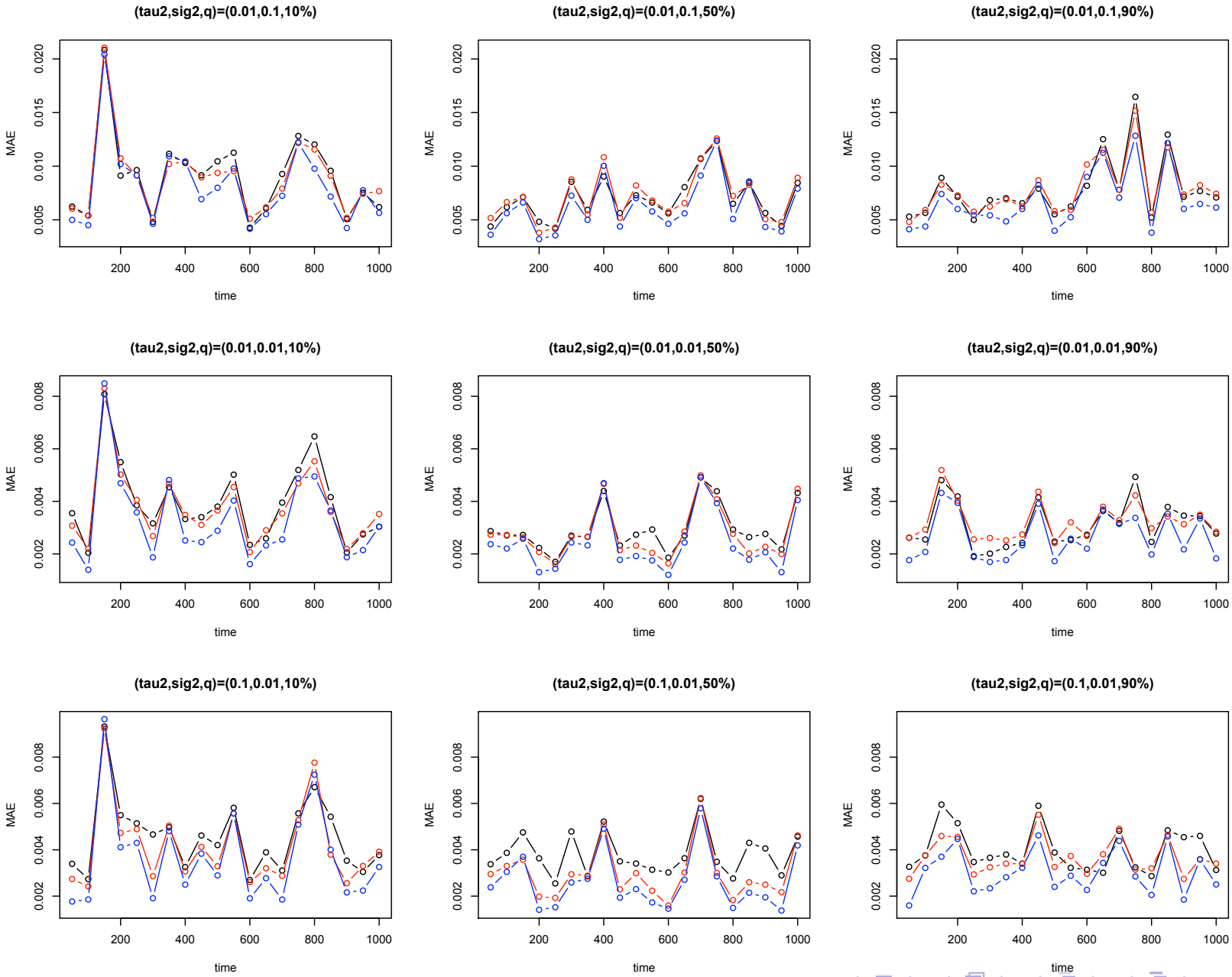
Example iii. $M = 5000$ and learning τ^2.



Example iii. $M = 500$ and learning x_t.



Example iii. $M = 5000$ and learning x_t.



Example iv. Computing sequential Bayes factors

A time series y_t is simulated from a *AR(1) plus noise* model:

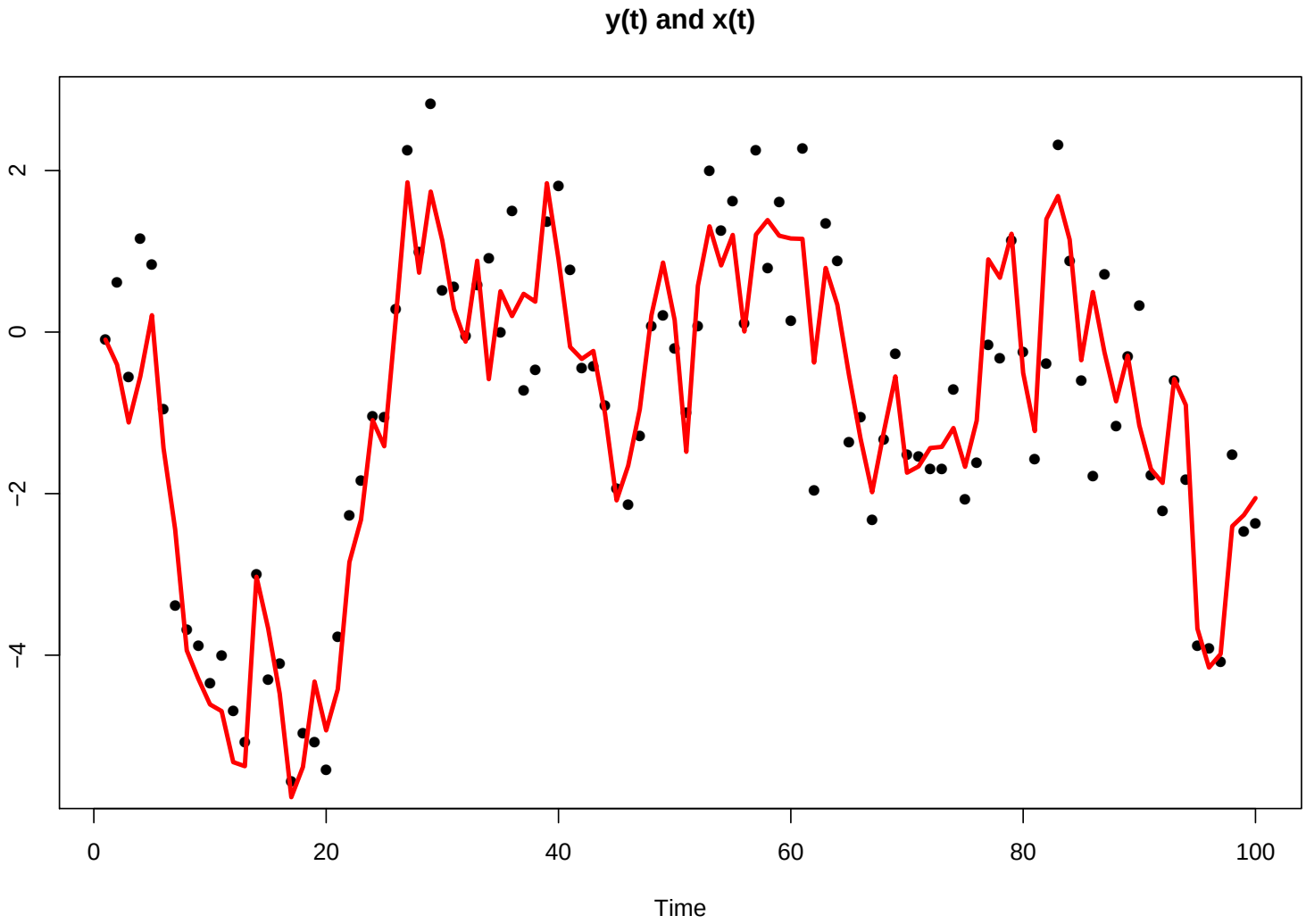
$$\begin{aligned}(y_{t+1}|x_{t+1},\theta) &\sim N(x_{t+1},\sigma^2)\\ (x_{t+1}|x_t,\theta) &\sim N(\beta x_t,\tau^2)\end{aligned}$$

for $t = 1,\ldots,T$.

We set $T = 100$, $x_0 = 0$, $\theta = (\beta,\sigma^2,\tau^2) = (0.9, 1.0, 0.5)$.

σ^2 and τ^2 are kept known and the independent prior distributions for β and x_0 are both $N(0,1)$.

Example iv. Simulated data



Example iv. PL pure filter versus PL

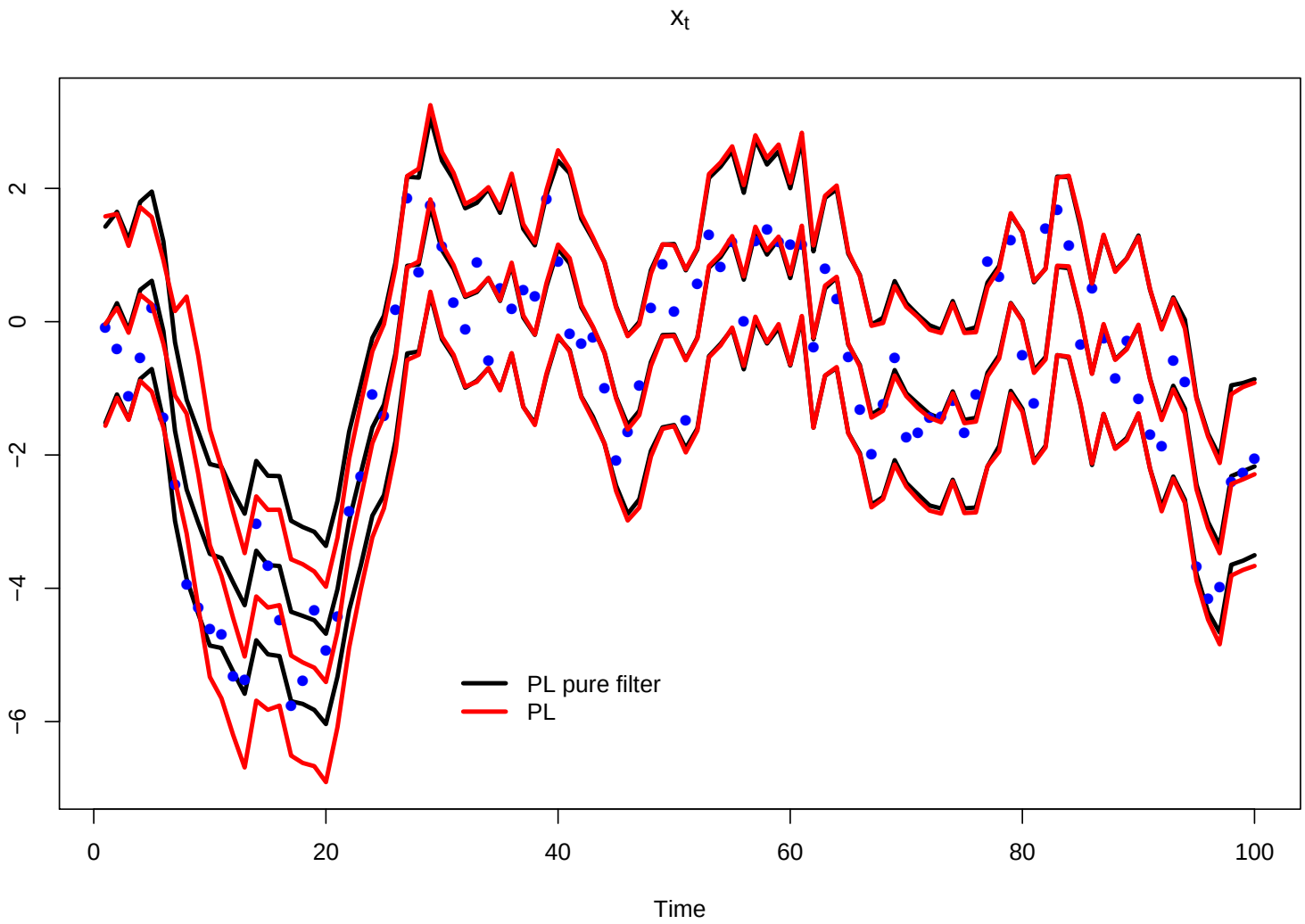
We run two filters:

- PL pure filter - our particle learning algorithm for learning x_t and keeping β fixed;
- PL - our particle learning algorithm for learning x_t and β sequentially.

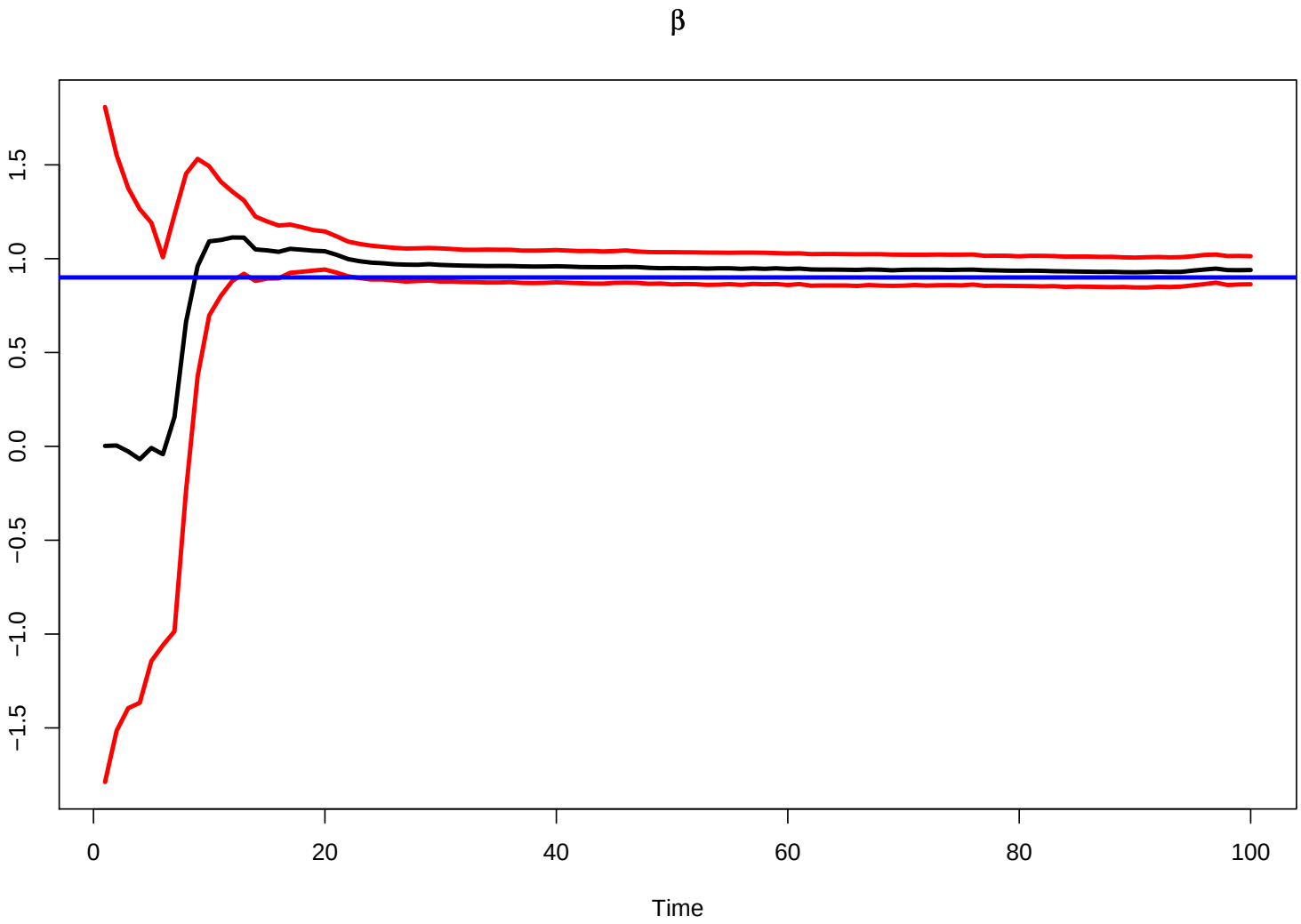
The filters are based on $N = 10,000$ particles.

Example iv. PL pure filter versus PL

β was fixed at the true value.

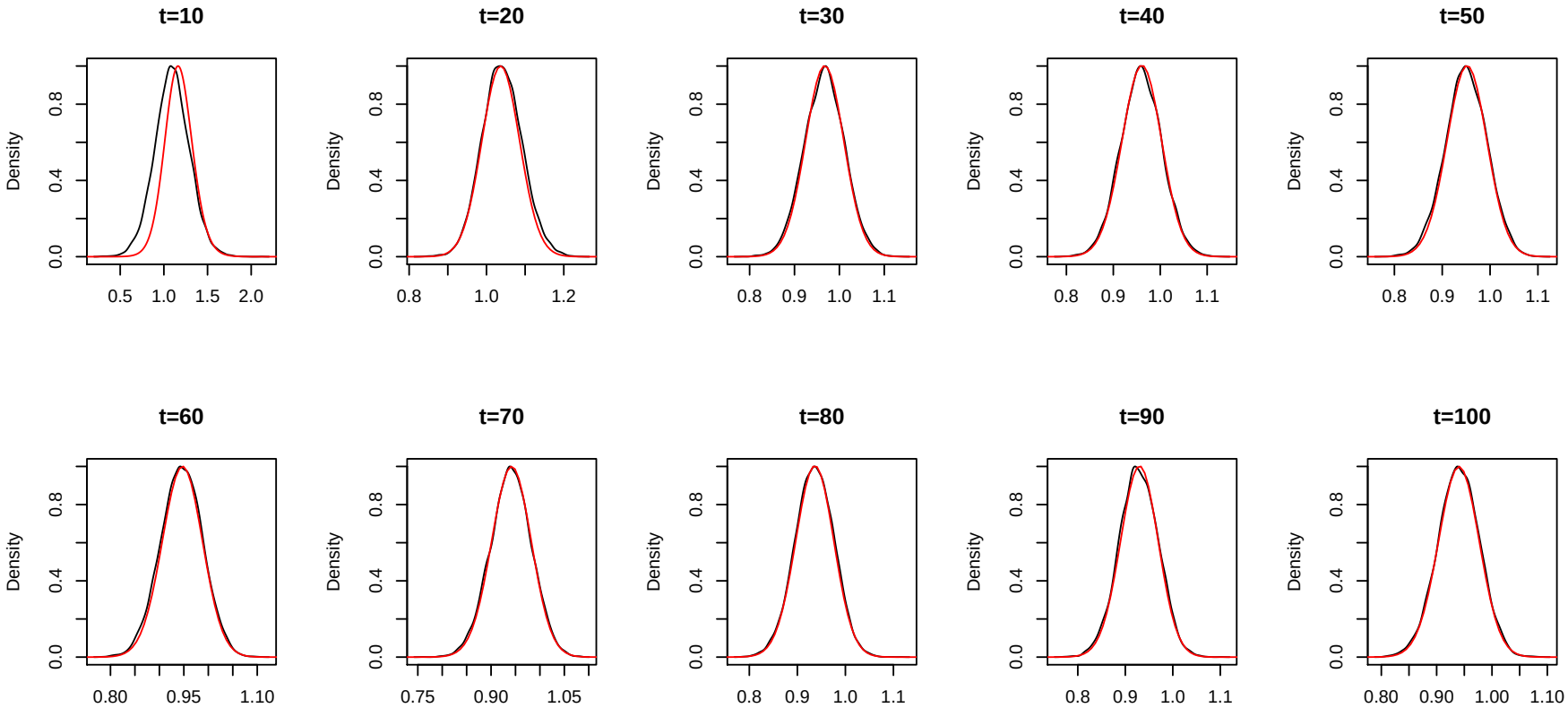


Example iv. PL - learning β

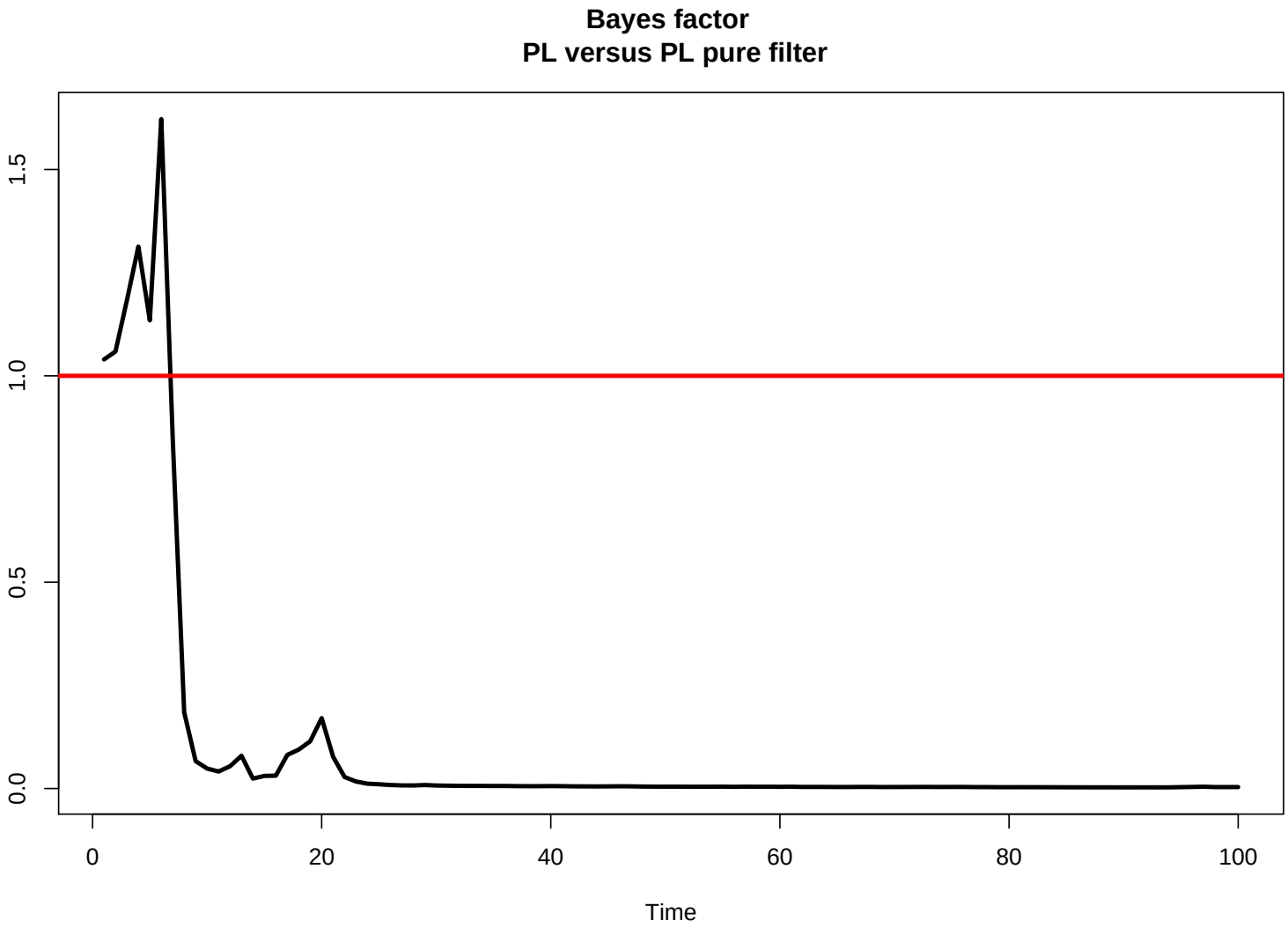


Example iv. PL - learning β

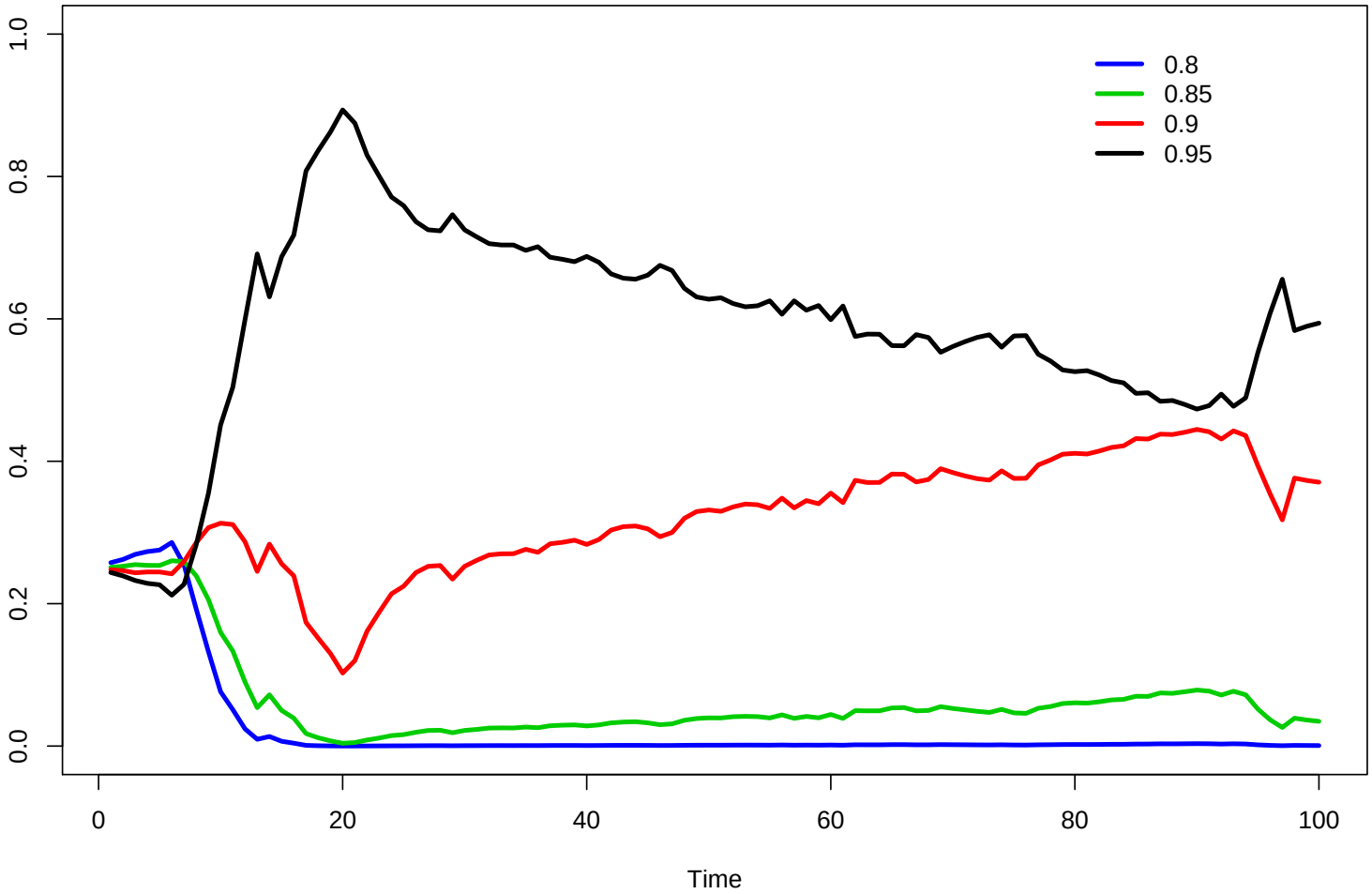
Comparing $p^N(\beta|y^t)$ with true $p(\beta|y^t)$.



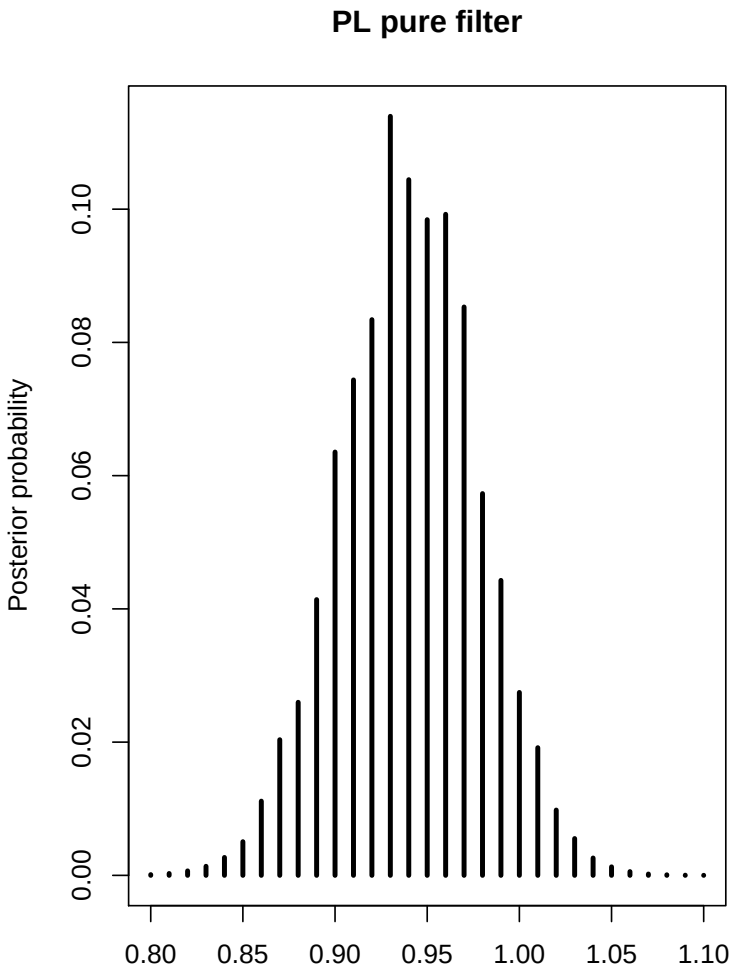
Example iv. Sequential Bayes factor

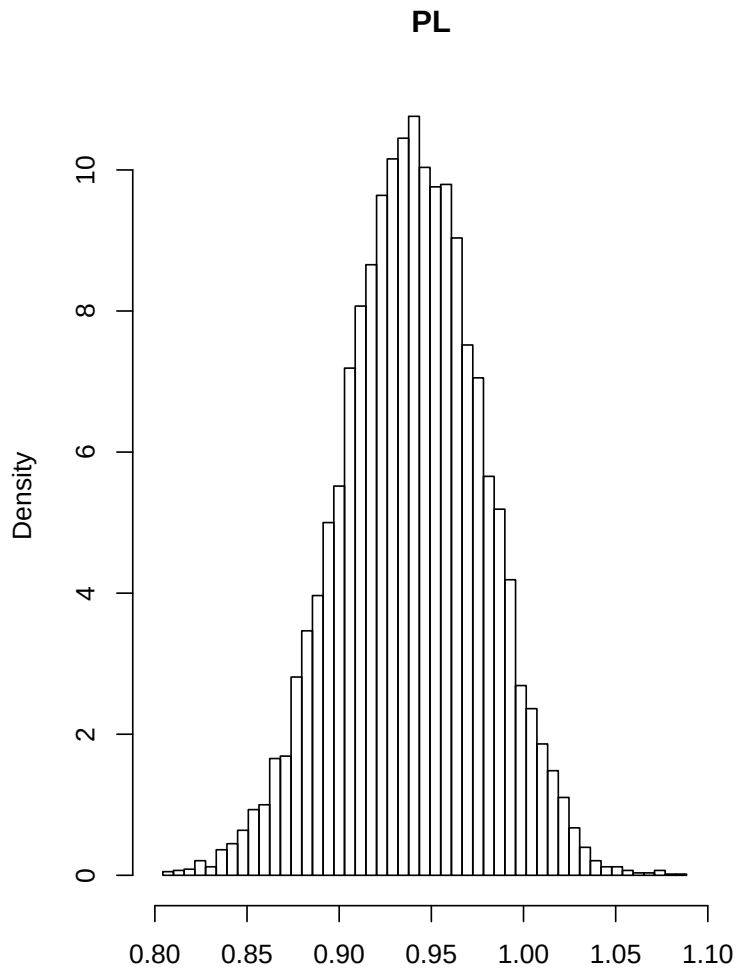


Example iv. Posterior model probabilities: 4 models



Example iv. Posterior model probabilities: 31 models





PL in Conditional Dynamic Linear Models (CDLM)

The model is

$$y_{t+1} = F_{\lambda_{t+1}} x_{t+1} + \epsilon_{t+1} \quad \text{where} \quad \epsilon_{t+1} \sim \mathcal{N}\left(0, V_{\lambda_{t+1}}\right)$$
$$x_{t+1} = G_{\lambda_{t+1}} x_t + \epsilon^x_{t+1} \quad \text{where} \quad \epsilon^x_{t+1} \sim \mathcal{N}\left(0, W_{\lambda_{t+1}}\right)$$

The error distribution

$$p\left(\epsilon_{t+1}\right) = \int \mathcal{N}\left(0, V_{\lambda_{t+1}}\right) p\left(\lambda_{t+1}\right) d\lambda_{t+1}$$

The augmented latent state is

$$\lambda_{t+1} \sim p(\lambda_{t+1}|\lambda_t)$$

PL extends Liu and Chen's (2000) "Mixture of Kalman Filters".

Algorithm

Step 1 (Re-sample): Generate an index $k(i) \sim Multi\left(w^{(i)}\right)$ where

$$w^{(i)} \propto p\left(y_{t+1}|(s_t^x, \theta)^{(i)}\right)$$

Step 2 (Propagate): States

$$\lambda_{t+1} \sim p\left(\lambda_{t+1}|\left(\lambda_t, \theta\right)^{k(i)}, y_{t+1}\right)$$

$$x_{t+1} \sim p(x_{t+1}|\left(x_t, \theta\right)^{k(i)}, \lambda_{t+1}, y_{t+1})$$

Step 3 (Propagate): Sufficient Statistics

$$s_{t+1}^x = \mathcal{K}\left(s_t^x, \theta, \lambda_{t+1}, y_{t+1}\right)$$

$$s_{t+1} = \mathcal{S}\left(s_t, x_{t+1}, \lambda_{t+1}, y_{t+1}\right)$$

Example v. Dynamic factor with switching loadings

For $t = 1, \ldots, T$, the model is defined as follows:

- Observation equation

$$y_t | z_t, \theta \sim N(\gamma_t x_t, \sigma^2 I_2)$$

- State equations

$$\begin{aligned} x_t | x_{t-1}, \theta &\sim N(x_{t-1}, \sigma_x^2) \\ \lambda_t | \lambda_{t-1}, \theta &\sim Ber((1-p)^{1-\lambda_{t-1}} q^{\lambda_{t-1}}) \end{aligned}$$

where $z_t = (x_t, \lambda_t)'$.

Factor loadings: $\gamma_t = (1, \beta_{\lambda_t})'$.

Parameters: $\theta = (\beta_1, \beta_2, \sigma^2, \sigma_x^2, p, q)'$.

Example v. Conditionally conjugate prior

$$
\begin{aligned}
(\beta_i|\sigma^2) &\sim N\left(b_{i0}, \sigma^2 B_{i0}\right) \qquad \text{for } i = 1, 2, \\
\sigma^2 &\sim IG\left(\frac{\nu_{00}}{2}, \frac{d_{00}}{2}\right) \\
\sigma_x^2 &\sim IG\left(\frac{\nu_{10}}{2}, \frac{d_{10}}{2}\right) \\
p &\sim Beta(p_1, p_2) \\
q &\sim Beta(q_1, q_2) \\
x_0 &\sim N(m_0, C_0)
\end{aligned}
$$

Example v. Particle representation

At time t, particles

$$\left\{ \left(x_t, \lambda_t, \theta, s_t^x, s_t \right)^{(i)} \right\}_{i=1}^{N}$$

approximating

$$p\left(x_t, \lambda_t, \theta, s_t^x, s_t | y^t \right)$$

where

- $s_t^x = \mathcal{S}(s_{t-1}^x, \theta)$ are state sufficient statistics
- $s_t = \mathcal{S}(s_{t-1}, x_t, \lambda_t)$ are fixed parameter sufficient statistics

Example v. Re-sampling $(x_t, \lambda_t, \theta, s_t^x, s_t)$

Let us redefine $\beta_i = (1, \beta_i)'$ whenever necessary.

Draw an index $k(i) \sim Multi(\omega^{(i)})$ with weights

$$\omega^{(i)} \propto p(y_{t+1}|(s_t^x, \lambda_t, \theta)^{k(i)})$$

with

$$p(y_{t+1}|m_t, C_t, \lambda_t, \theta) = \sum_{j=1}^{2} f_N\left(y_{t+1}; \beta_j m_t, V_j\right) Pr\left(\lambda_{t+1} = j|\lambda_t, \theta\right)$$

where $V_j = (C_t + \sigma_x^2)\beta_j\beta_j' + \sigma^2 I_2$, m_t and C_t are components of s_t^x and f_N denotes the normal density function.

Example v. Propagating states

Draw auxiliary state λ_{t+1}

$$\lambda_{t+1}^{(i)} \sim p(\lambda_{t+1}|(s_t^x, \lambda_t, \theta)^{k(i)}, y_{t+1})$$

where

$$Pr(\lambda_{t+1} = j|s_t^x, \lambda_t, \theta, y_{t+1}) \propto f_N\left(y_{t+1}; \beta_j m_t, V_j\right) p\left(\lambda_{t+1} = j|\lambda_t, \theta\right).$$

Draw state x_{t+1} conditionally on λ_{t+1}

$$x_{t+1}^{(i)} \sim p(x_{t+1}|\lambda_{t+1}^{(i)}, (s_t^x, \theta)^{k(i)}, y_{t+1})$$

by a simply Kalman filter update.

Example v. Updating sufficient statistics for states, s^x_{t+1}

The Kalman filter recursion yield

$$m_{t+1} = m_t + A_{t+1}(y_{t+1} - \beta_{\lambda_{t+1}} m_t)$$
$$C_{t+1} = C_t + \sigma_x^2 - A_{t+1} Q_{t+1}^{-1} A'_{t+1}$$

where

$$Q_{t+1} = (C_t + \sigma_x^2)\gamma_{t+1}\gamma'_{t+1} + \sigma^2 I_2$$
$$A_{t+1} = (C_t + \sigma_x^2)\gamma'_{t+1} Q_{t+1}^{-1}$$

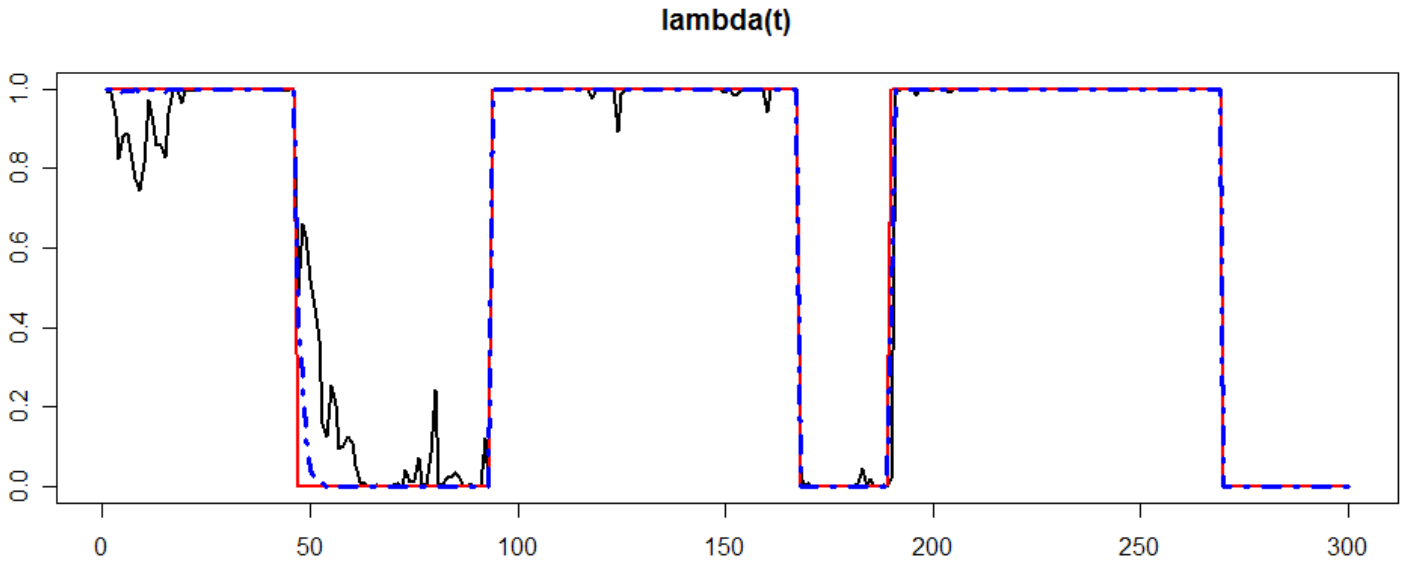
Example v. Updating suff. statistics for parameters, s_{t+1}

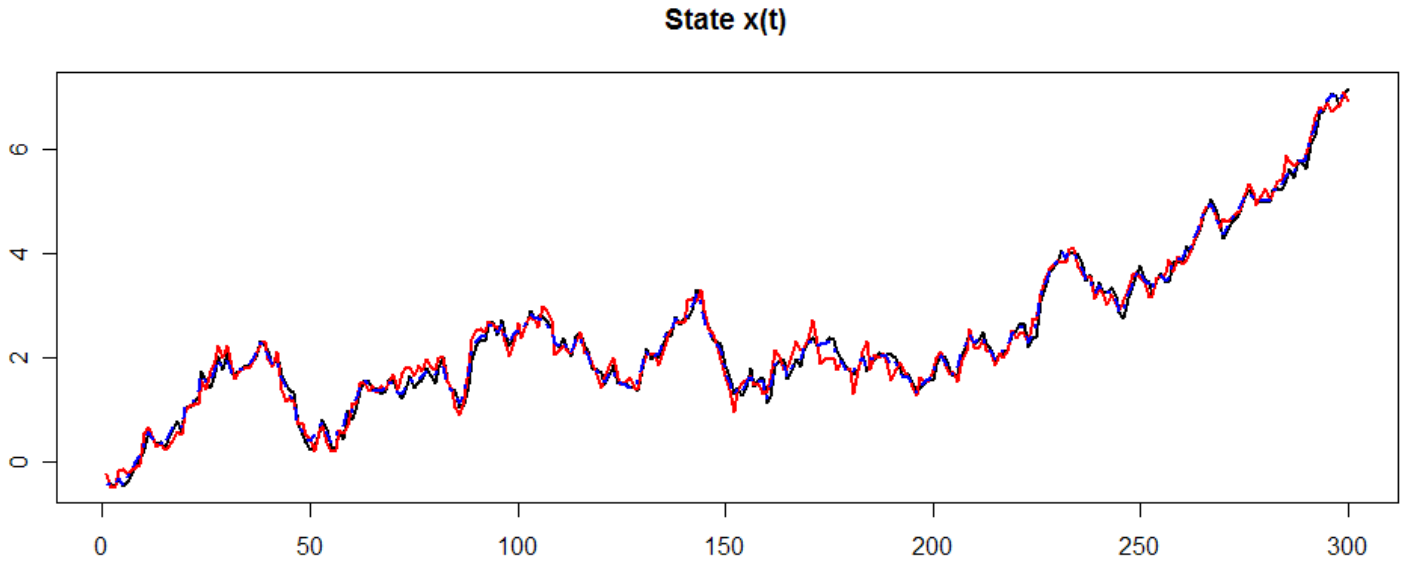
Recall that $s_{t+1} = \mathcal{S}(s_t, x_{t+1}, \lambda_{t+1})$. Then,

$$
\begin{aligned}
(\beta_i|\sigma^2, s_{t+1}) &\sim N\left(b_{i,t+1}, \sigma^2 B_{i,t+1}\right) \qquad \text{for } i = 1, 2, \\
(\sigma^2|s_{t+1}) &\sim IG\left(\frac{\nu_{0t}}{2}, \frac{d_{0,t+1}}{2}\right) \\
(\sigma_x^2|s_{t+1}) &\sim IG\left(\frac{\nu_{1t}}{2}, \frac{d_{1,t+1}}{2}\right) \\
(p|s_{t+1}) &\sim Beta(p_{1,t+1}, p_{2,t+1}) \\
(q|s_{t+1}) &\sim Beta(q_{1,t+1}, q_{2,t+1})
\end{aligned}
$$

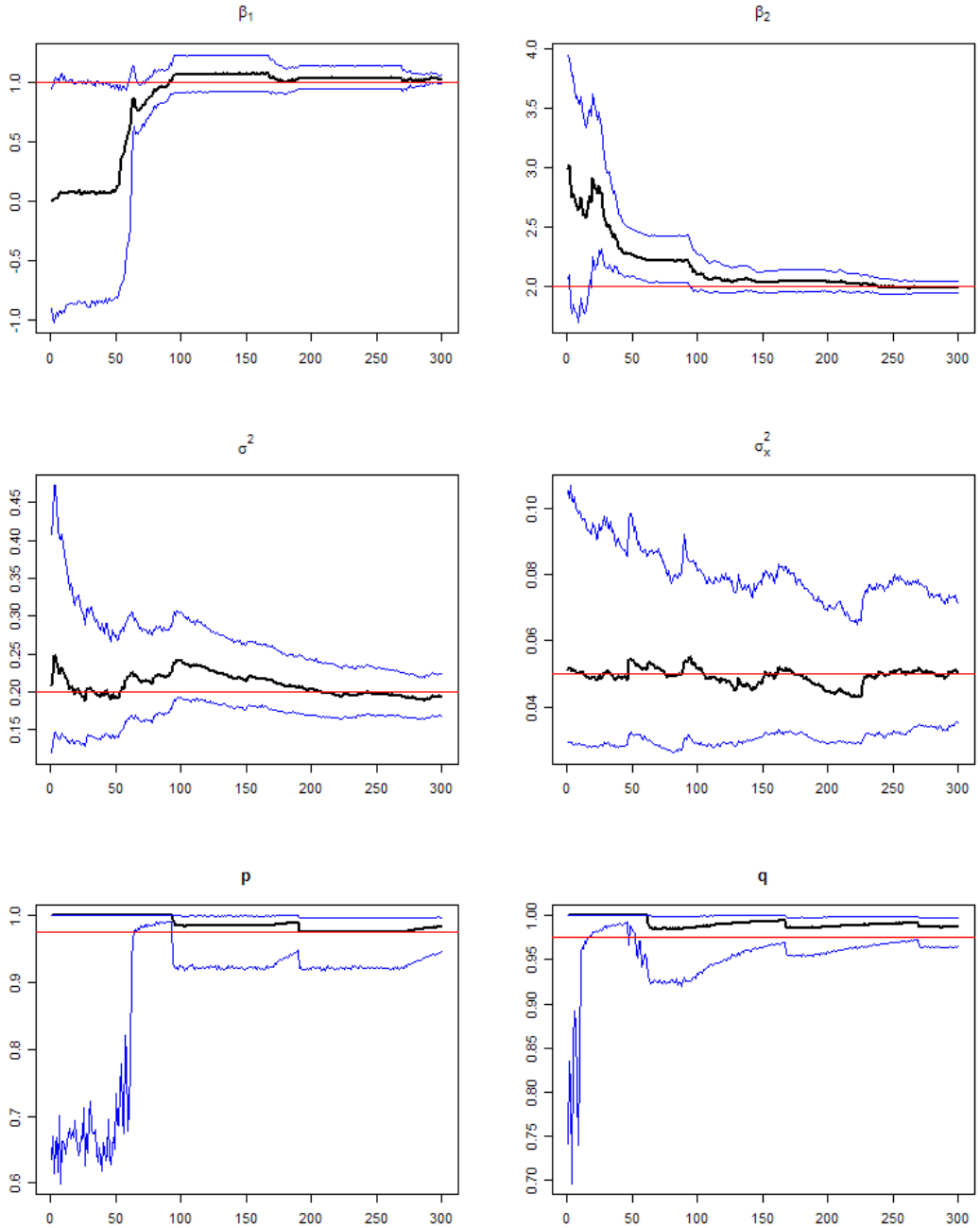
where $\mathbb{I}_{\lambda_{t+1}=i} = \mathbb{I}_i$, $\mathbb{I}_{\lambda_t=i,\lambda_{t+1}=j} = \mathbb{I}_{ij}$, $\nu_{it} = \nu_{i,t-1} + 1$,
$B_{i,t+1}^{-1} = B_{it}^{-1} + x_{t+1}^2$, $B_{i,t+1}^{-1} b_{i,t+1} = B_{it}^{-1} b_{it} + x_{t+1} y_{t+1,2} \mathbb{I}_i$,
$p_{i,t+1} = p_{it} + \mathbb{I}_{1i}$ (similarly for $q_{i,t+1}$) for $i = 1, 2$,
$d_{0,t+1} = d_{0,t} + (y_{t+1,1} - x_{t+1})^2 +$
$\sum_{j=1}^{2} \left[(y_{t+1,2} - b_{j,t+1} x_{t+1}) y_{t+1,2} + B_{j,t+1}^{-1} b_{j,t+1}\right] \mathbb{I}_j$, and
$d_{1,t+1} = d_{1,t} + (x_{t+1} - x_t)^2$.

Example v. Filtering and smoothing for states





Example v. Sequential parameter learning



PL in (state) non-linear normal dynamic models

The model now is

$$y_{t+1} = F_{\lambda_{t+1}} x_{t+1} + \epsilon_{t+1} \quad \text{where} \quad \epsilon_{t+1} \sim \mathcal{N}\left(0, V_{\lambda_{t+1}}\right)$$
$$x_{t+1} = G_{\lambda_{t+1}} Z\left(x_t\right) + \omega_{t+1} \quad \text{where} \quad \omega_{t+1} \sim \mathcal{N}\left(0, W_{\lambda_{t+1}}\right)$$

where ϵ_{t+1} and λ_{t+1} are modeled as before.

Algorithm:

Step 1 (Re-sample): Generate an index $k(i) \sim Multi\left(w^{(i)}\right)$ where

$$w^{(i)} \propto p\left(y_{t+1} | (x_t, \theta)^{(i)}\right)$$

Step 2 (Propagate):

$$\lambda_{t+1} \sim p\left(\lambda_{t+1} | \left(\lambda_t, \theta\right)^{k(i)}, y_{t+1}\right)$$
$$x_{t+1} \sim p\left(x_{t+1} | \left(x_t, , \theta\right)^{k(i)}, \lambda_{t+1}, y_{t+1}\right)$$
$$s_{t+1} = \mathcal{S}\left(s_t, x_{t+1}, \lambda_{t+1}, y_{t+1}\right)$$

Example vi. Fat-tailed nonlinear model

Let

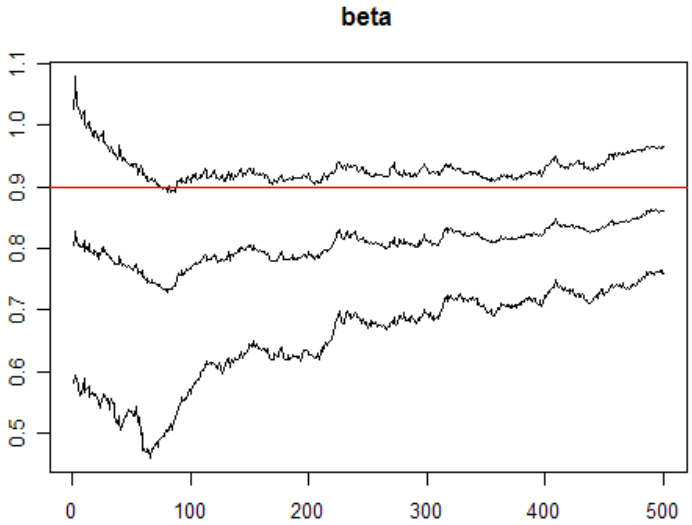
$$y_{t+1} = x_{t+1} + \sigma\sqrt{\lambda_{t+1}}\epsilon_{t+1} \quad \text{where} \quad \lambda_{t+1} \sim \mathcal{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$$

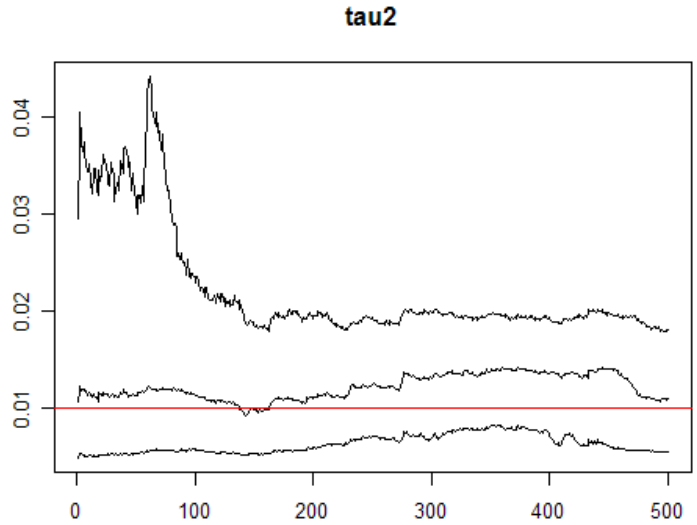
$$x_{t+1} = g(x_t)\beta + \sigma_x u_{t+1} \qquad \text{where} \quad g(x_t) = \frac{x_t}{1 + x_t^2}$$

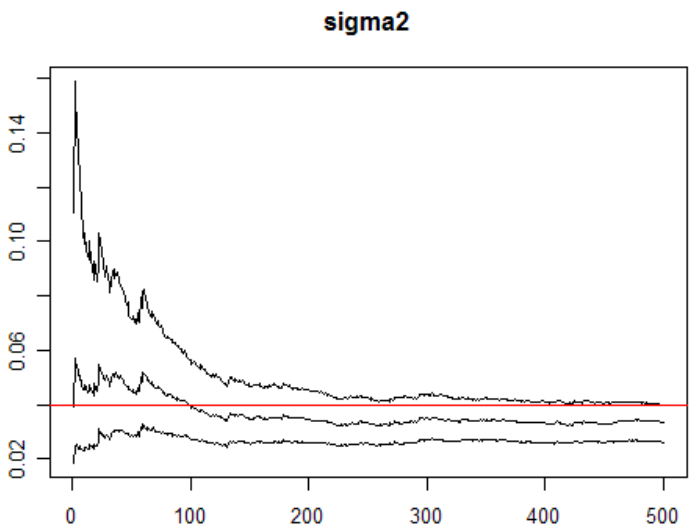
where ϵ_{t+1} and u_{t+1} are independent standard normals and ν is known.

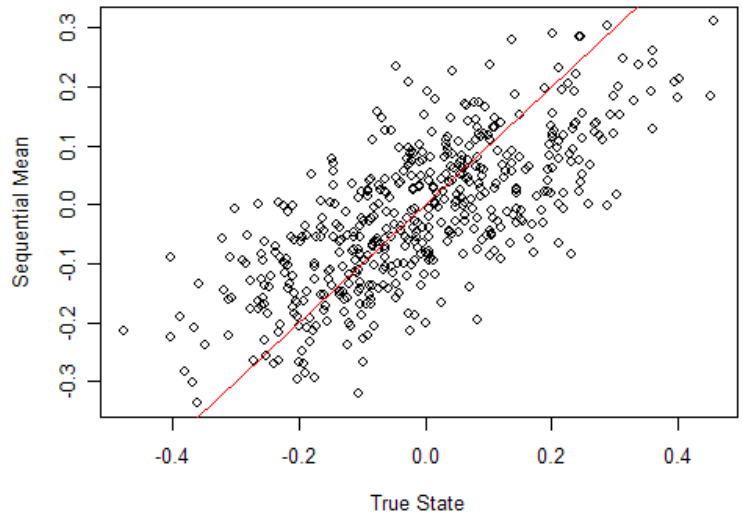
The observation error term is non-normal $\sqrt{\lambda_{t+1}}\epsilon_{t+1} \sim t_\nu$.

Example vi. Sequential inference









Example vii. Dynamic multinomial logit model

Let us study the multinomial logit model

$$P\left(y_{t+1}=1|\beta_{t+1}\right)=\frac{e^{F_t\beta_t}}{1+e^{F_t\beta_t}} \quad \text{and} \quad \beta_{t+1} \quad =\phi\beta_t+\sigma_x\epsilon_{t+1}^{\beta}$$

where $\beta_0 \sim N(0, \sigma^2/(1-\rho^2))$. Scott's (2007) data augmentation structure leads to a mixture Kalman filter model

$$\begin{aligned} y_{t+1} &= \mathbb{I}(z_t \geq 0) \\ z_{t+1} &= Z_t\beta+\epsilon_{t+1} \ \text{ where } \ \epsilon_{t+1} \sim -ln\mathcal{E}(1) \end{aligned}$$

Here ϵ_t is an extreme value distribution of type 1 where $\mathcal{E}(1)$ is an exponential of mean one. The key is that it is easily to simulate $p(z_t|\beta, y_t)$ using

$$z_{t+1}=-\ln\left(\frac{\ln U_i}{1+e^{\beta_i\beta}}-\frac{\ln V_i}{e^{\beta_i\beta}}\mathcal{I}_{y_{t+1}=0}\right)$$

Example vii. 10-component mixture of normals

Frunwirth-Schnatter and Schnatter (2007) uses a 10-component mixture of normals:

$$p(\epsilon_t) = e^{-\epsilon_t - e^{-\epsilon_t}} \approx \sum_{j=1}^{10} w_j \mathcal{N}(\mu_j, s_j^2)$$

Hence conditional on an indicator λ_t we can analyze

$$y_t = \mathbb{I}(z_t \geq 0) \quad \text{and} \quad z_t = \mu_{\lambda_t} + Z_t\beta + s_{\lambda_t}\epsilon_t$$

where $\epsilon_t \sim N(0,1)$ and $Pr(\lambda_t = j) = w_j$. Also,

$$\begin{aligned} s_{t+1}^{\beta} &= \mathcal{K}\left(s_t^{\beta}, z_{t+1}, \lambda_{t+1}, \theta, y_{t+1}\right) \\ p(y_{t+1}|s_t^{\beta}, \theta) &= \sum_{\lambda_{t+1}} p(y_{t+1}|s_t^{\beta}, \lambda_{t+1}, \theta) \end{aligned}$$

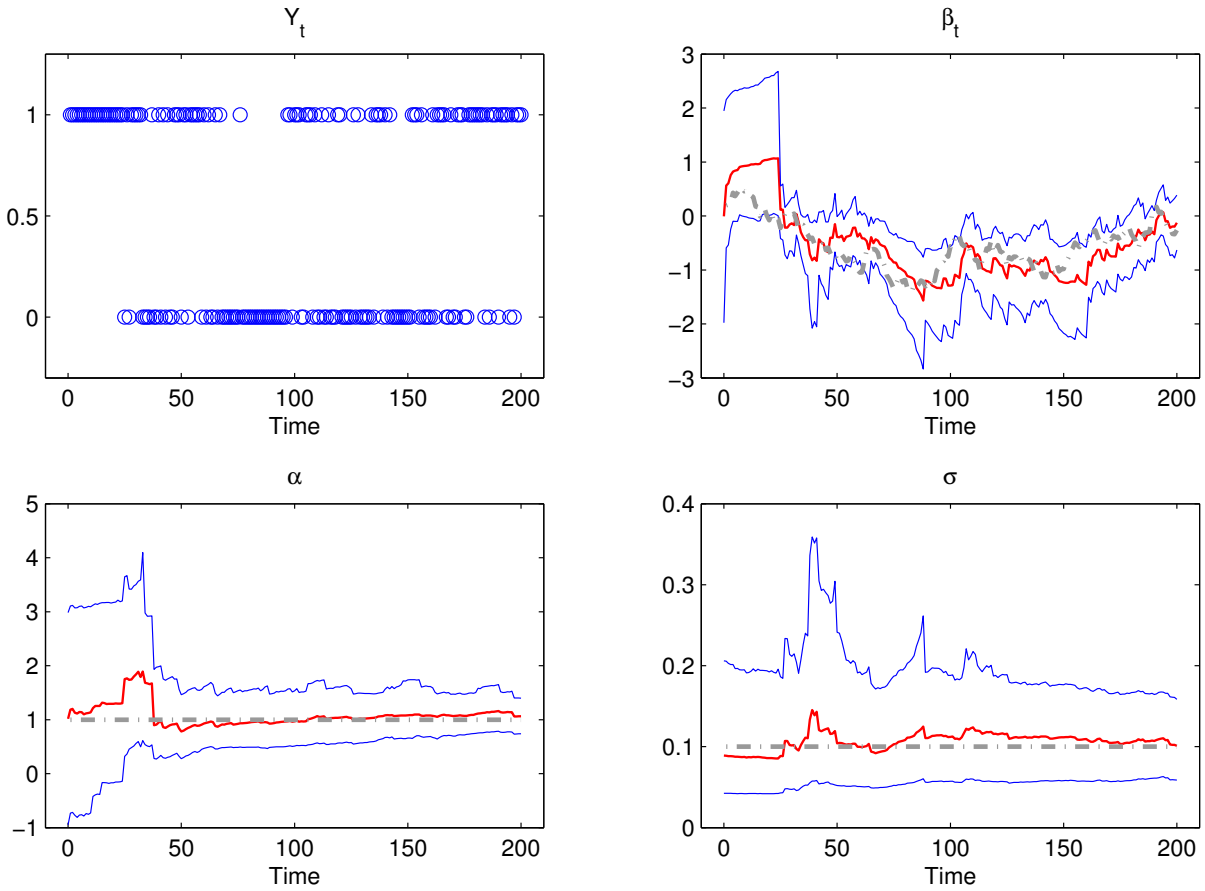
for re-sampling. Propagation now requires

$$\begin{aligned} \lambda_{t+1} &\sim p\left(\lambda_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, y_{t+1}\right) \\ z_{t+1} &\sim p\left(z_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, \lambda_{t+1}, y_{t+1}\right) \\ \beta_{t+1} &\sim p\left(z_{t+1}|(s_t^{\beta}, \theta)^{k(i)}, \lambda_{t+1}, z_{t+1}\right) \end{aligned}$$

where λ_{t+1} comes from a discrete distribution.

Followed by the deterministic updating for conditional sufficient statistics.

Example vii. Simulated exercise



PL based on 30,000 particles.

Example viii. Sequential Bayesian Lasso

We develop a sequential version of Bayesian Lasso[2] for a simple problem of signal detection. The model takes the form

$$\begin{aligned}(y_t|\theta_t) &\sim N(\theta_t, 1)\\ p(\theta_t|\tau) &= (2\tau)^{-1}\exp\left(-|\theta_t|/\tau\right)\end{aligned}$$

for $t = 1, \ldots, n$ and $\tau^2 \sim IG(a_0, b_0)$.

Data augmentation: It is easy to see that

$$p(\theta_t|\tau) = \int p(\theta_t|\tau, \lambda_t)p(\lambda_t)d\lambda_t$$

where

$$\begin{aligned}\lambda_t &\sim Exp(2)\\ \theta_t|\tau, \lambda_t &\sim N(0, \tau^2\lambda_t)\end{aligned}$$

[2] Carlin and Polson (1991) and Hans (2009)

Example viii. Data augmentation

The natural set of latent variables is given by the augmentation variable λ_{n+1} and conditional sufficient statistics leading to

$$Z_n = (\lambda_{n+1}, a_n, b_n)$$

The sequence of variables λ_{n+1} are i.i.d. and so can be propagated directly with $p(\lambda_{n+1})$.

The conditional sufficient statistics (a_{n+1}, b_{n+1}) are deterministically determined based on parameters $(\theta_{n+1}, \lambda_{n+1})$ and previous values (a_n, b_n).

Example viii. PL algorithm

1. After n observations: $\left\{(Z_n, \tau)^{(i)}\right\}_{i=1}^{N}$.
2. Draw $\lambda_{n+1}^{(i)} \sim Exp(2)$.
3. Resample old particles with weights

$$w_{n+1}^{(i)} \propto p(y_{n+1}; 0, 1+\tau^{2(i)}\lambda_{n+1}^{(i)}).$$

4. Sample $\theta_{n+1}^{(i)} \sim N(m_n^{(i)}, C_n^{(i)})$, where $m_n^{(i)} = C_n^{(i)} y_{n+1}$ and $C_n^{-1} = 1 + \tilde{\tau}^{-2(i)}\tilde{\lambda}_{n+1}^{-1(i)}$.
5. Suff. stats: $a_{n+1}^{(i)} = \tilde{a}_n^{(i)} + 1/2$, $b_{n+1}^{(i)} = \tilde{b}_n^{(i)} + \theta_{n+1}^{2(i)}/(2\tilde{\lambda}_{n+1}^{(i)})$.
6. Sample (offline) $\tau^{2(i)} \sim IG(a_{n+1}, b_{n+1})$.
7. Let $Z_{n+1}^{(i)} = (\lambda_{n+1}^{(i)}, a_{n+1}^{(i)}, b_{n+1}^{(i)})$.
8. After $n+1$ observations: $\left\{(Z_{n+1}, \tau)^{(i)}\right\}_{i=1}^{N}$.

Example viii. Sequential Bayes factor

As the Lasso is a model for sparsity we would expect the evidence for it to increase when we observe $y_t = 0$.

We can sequentially estimate $p(y_{n+1} \mid y^n, \text{lasso})$ via

$$p(y_{n+1} \mid y^n, \text{lasso}) = \frac{1}{N} \sum_{i=1}^{N} p(y_{n+1} \mid (\lambda_n, \tau)^{(i)})$$

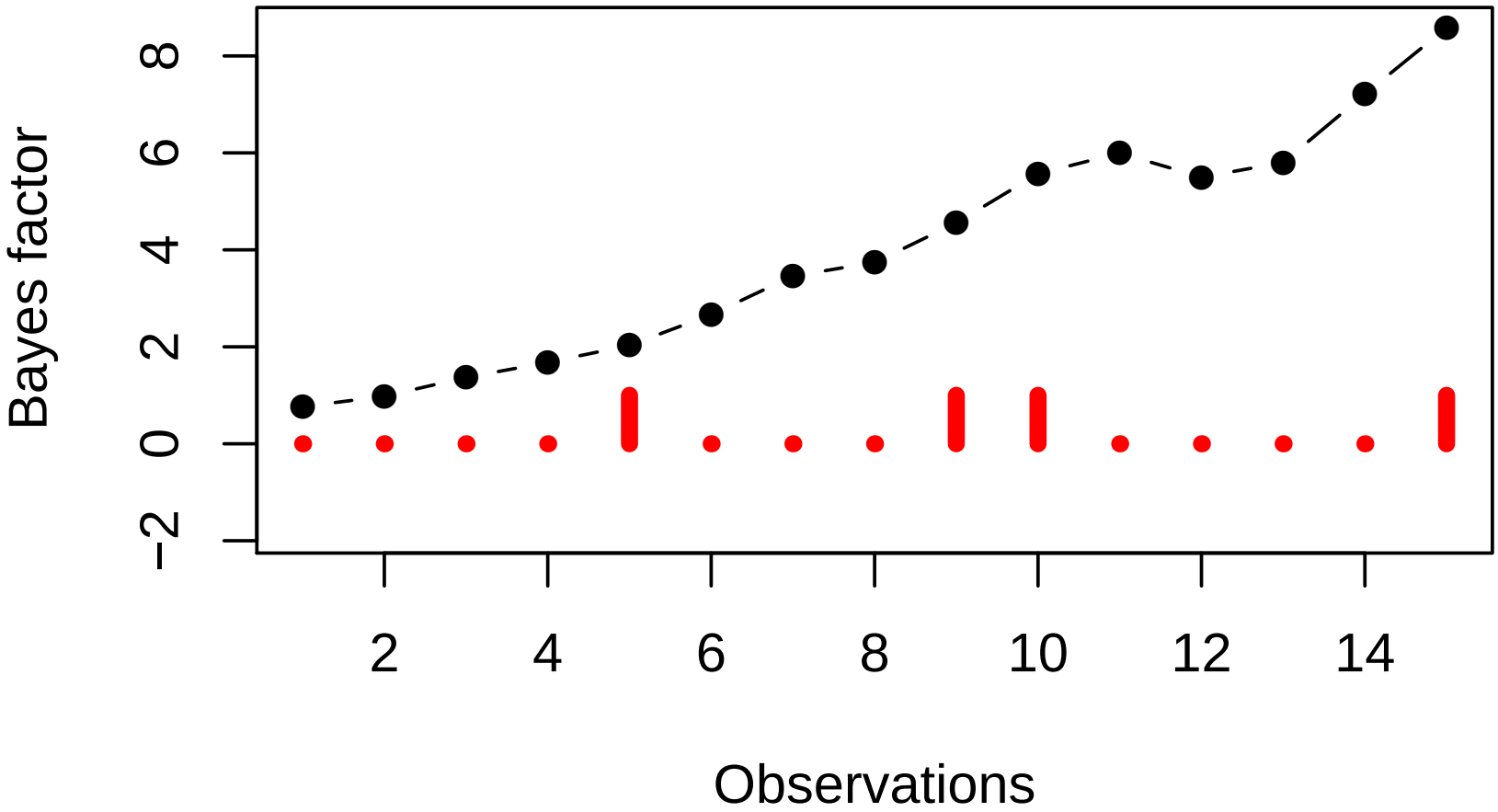
with predictive $p(y_{n+1} \mid \lambda_n, \tau) \sim N(0, \tau^2 \lambda_n + 1)$.

This leads to a sequential Bayes factor

$$BF_{n+1} = \frac{p(y^{n+1} \mid \text{lasso})}{p(y^{n+1} \mid \text{normal})}.$$

Example viii. Simulated data

Data based on $\theta = (0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1)$ and priors $\tau^2 \sim IG(2, 1)$ for the double exponential case and $\tau^2 \sim IG(2, 3)$ for the normal case, reflecting the ratio of variances between those two distributions.



Final remarks

PL is a general framework for sequential Bayesian inference in dynamic and static models.

PL is able to deal with filtering and learning and reduce the accumulation of error.

The loose definition of sufficient statistics and the flexibility to freely augment x_t makes PL a competitive alternative to MCMC in highly structured models.

A powerful by-product of PL (and SMC in general) over MCMC schemes, is its ability to sequentially produce model comparison, assessment indicators.

Basic references

Briers, Doucet and Maskell (2010) Smoothing algorithms for state-space models. Annals of the Institute Statistical Mathematics, 62, 61-89.

Carlin and Polson (1991). Inference for Nonconjugate Bayesian Models Using the Gibbs Sampler. The Canadian Journal of Statistics, 19, 399-405.

Carvalho, Johannes, Lopes and Polson (2010) Particle learning and smoothing, Statistical Science, 25, 88-106.

Chen and Liu (2000). Mixture Kalman filter. Journal of the Royal Statistical Society, Series B, 62, 493-508.

Fearnhead (2002). Markov chain Monte Carlo, sufficient statistics and particle filter. Journal of Computational and Graphical Statistics, 11, 848-62.

Fearnhead, Wyncoll and Tawn (2010) A sequential smoothing algorithm with linear computational cost. Biometrika, 97,447-464.

Frühwirth-Schnatter and Frühwirth (2007) Auxiliary mixture sampling with applications to logistic models. Computational Statistics and Data Analysis, 51, 3509-3528.

Godsill SJ, Doucet A, West M. 2004. Monte Carlo smoothing for non-linear time series. Journal of the American Statistical Association, 99, 156-168.

Gordon, Salmond and Smith (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F 140*, 107-113.

Hans (2009) Bayesian lasso regression. Biometrika, 96, 835-845.

Kong, Liu and Wong (1994). Sequential imputation and Bayesian missing data problems. Journal of the American Statistical Association, 89, 590-99.

Liu and West (2001) Combined parameters and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice* (Eds. A. Doucet, N. de Freitas and N. Gordon). New York: Springer-Verla, 197-223.

Lopes, Carvalho, Johannes and Polson (2011) Particle learning for sequential Bayesian computation (with discussion), Bayesian Statistics 9, 2011, 317-360.

Lopes and Tsay (2011) Particle filters and Bayesian inference in financial econometrics, Journal of Forecasting, 30, 168-209.

Pitt and Shephard (1999) Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94, 590-599.

Prado and West (2010) *Time Series: Modelling, Computation and Inference*. Baton Rouge: Chapman & Hall/CRC.

Storvik (2002) Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions of Signal Processing*, 50, 281-289.